

Getting Ready for the New Millennium

Year **2000** Remediation
for Information Builders Products

Version 1.0

Assembled by the CTSS Support Technology Group

Information Builders

Corporate Headquarters Two Penn Plaza, New York, NY 10121-2898 (212) 736-4433 Fax (212) 967-6406
World Wide Web: www.ibi.com E-mail: info@ibi.com

Canadian Headquarters 150 York St., Suite 1000, Toronto, ON M5H 3S5 (416) 364-2760 Fax (416) 364-6552

For International Inquiries Fax: +1(212) 279-2382

Copyright © 1999 by Information Builders, Inc. All rights reserved.

All products and product names mentioned in this publication are trademarks or registered trademarks of their respective companies.

Contents

1	Introduction	1-1
	Feedback	1-2
	Updates	1-2
	Acknowledgements	1-2
	Year 2000 Basics.....	1-3
	Applications	1-3
	Products.....	1-4
	Remediation	1-5
	Summary.....	1-5
	Year 2000 Compliance Issues Statement	1-6
	Background.....	1-6
	Criteria Framework.....	1-7
	Table 1: Year 2000 Date Integrity Criteria	1-7
	Table 2: Integrity Capability for Applications Using Full Date Containers.....	1-8
	Other Products	1-11
	Other Date Representation Methods	1-11
	Table 3: Remediation Features for Applications Not Using Full Date Containers	1-11
	Application Remediation	1-12
	NOTICE.....	1-12
2	Remediation in FOCUS	2-1
	Determining If You Have a Year 2000 Problem.....	2-2
	Data Display.....	2-3
	Data Entry	2-3
	Data Processing.....	2-4
	Data Storage.....	2-4
	Smart Dates in FOCUS files	2-5
	Smart Dates With the Century	2-7
	Smart Dates Without the Century	2-9
	Smart Dates in Non-FOCUS Files	2-10
	Tools Available:	2-12
	Using FOCUS Code to Scan for Date Fields	2-12
	SiteAnalyzer.....	2-13
	WALDO.....	2-14
	Professional Services	2-14
	Systems Journal.....	2-15
3	FOCUS for IBM Mainframe	3-1
	What Is Remediation?	3-2
	What To Do If You Are In a Year 2000 Capable Release	3-2
	Sliding Window – DEFCENT and YRTHRESH.....	3-3
	DEFCENT, YRTHRESH and Maintain.....	3-5
	Temporary Field Level Example	3-5
	Field Level Example	3-5

Table of Contents

File Level Example	3-5
Global Level Example	3-6
Date Expansion	3-7
Date/Data Conversion	3-7
How to Reload the Database to Store the Correct Century	3-8
I6YMD Non-Key Field Conversion Process	3-8
I6YMD Key Field Conversion Process	3-10
A6YMD Non-Key Field Conversion Process	3-11
A6YMD Key Field Conversion Process	3-12
A6MDY AS A KEY FIELD	3-13
Changing the Century That Was Implicitly Stored With Smart Dates.....	3-14
Application Changes in FOCUS for IBM Mainframe	3-15
Converting From Traditional Dates to Smart Dates.....	3-16
What if you are not using Smart Dates?.....	3-17
User-Written Subroutines	3-17
Functions.....	3-18
Displaying the Date.....	3-19
Sorting.....	3-20
External Sorts.....	3-20
Selection Tests (on date fields)	3-21
Smart Dates	3-21
Traditional Dates.....	3-22
Relational (SQL) Files	3-23
Non-Relational (Non-SQL) Files	3-23
Dialogue Manager and &Variables.....	3-23
Working With Non-Year 2000 Capable Releases.....	3-24
What if DEFCENT and YRTHRESH are not available?.....	3-24
Using the DAxxx (Date) Subroutines:	3-25
Using Smart Dates	3-25
Limitations of FOCUS Date Handling.....	3-26
Testing on FOCUS for IBM Mainframe	3-27
Staffing.....	3-27
Environment.....	3-27
Data	3-28
Test Dates.....	3-29
Test Dates for Applications.....	3-29
Test Dates for Hardware	3-30
Test Dates for Systems Software	3-31
Testing Checklist.....	3-31
Testing Issues	3-32
Source Control and Management.....	3-32
Unit Testing Tasks	3-32
Business Cycle Testing	3-32
Test Coverage	3-33
Platform Specific Considerations.....	3-33
Other Considerations.....	3-33

Table of Contents

Summary of New Date Features	3-34
Release 7.0.5	3-34
Release 7.0.6	3-34
Release 7.0.7M.....	3-34
Release 7.0.8	3-35
Release 7.0.8R.....	3-35
Frequently Asked Questions	3-36
1. What is the Mega PTF?.....	3-36
2. Is the Mega PTF (PTF 87857 for MVS and PTF88008 for VM) that is required for 7.0.8 included in 7.0.8R, or is it also required for 7.0.8R?	3-36
3. Where can I find the different features and fixes that are in releases 7.0.8 and 7.0.8R, as well as other 7.0 releases?.....	3-37
4. What are DEFCENT and YRTHRESH?.....	3-37
5. How does setting DEFCENT and YRTHRESH affect using COMBINES in the application?.....	3-37
6. Are DEFCENT and YRTHRESH used in MODIFY procedures?.....	3-37
7. Does MAINTAIN support DEFCENT and YRTHRESH?.....	3-37
8. What field formats can I apply DEFCENT and YRTHRESH to?	3-37
9. What happens if you code DEFCENT and YRTHRESH, but the field already contains the century?	3-38
10. What does the 'R' stand for in release 7.0.8R?	3-38
11. Is the Cobol FD Translator (CTF) Year 2000 compliant?	3-38
12. Does the FOCUS Graph facility work with Year 2000 dates?.....	3-38
13. Can my application be Year 2000 compliant if I'm in a release prior to 7.0.8 and if I do not use any of the Date Subroutines?.....	3-38
14. Why do I get the wrong value returned when I DEFINE a field to the system date?	3-38
4 EDA	4-1
Determining If You Have a Year 2000 Problem	4-1
Determining Where to Remediate.....	4-2
When might remediation on the Client be necessary?	4-2
When is it better to remediate on the EDA Server machine:.....	4-2
Determining What to Remediate	4-3
Level 1: Application Considerations	4-3
Data Presentation Considerations:	4-4
How does EDA return date columns with and without a century?	4-4
SQL SELECT requests	4-4
SQL Scenario 1: DATE Datatype Columns in SQL Passthru Mode - All EDA Releases	4-5
SQL Scenario 2: Numeric Full Date Container Columns - All EDA Releases.....	4-6
SQL Scenario 3: Alphanumeric Full Date Container Columns - EDA Release 3.3.2 (MVS) & 4.x (All Platforms)	4-7
SQL Scenario 4: Alphanumeric Full Date Container Columns - EDA Release 3.3.1 & below.....	4-8

Table of Contents

SQL Scenario 5: Implicit Century Numeric
 Partial Date Container Columns - All EDA Releases 4-9

SQL Scenario 6: Legacy Numeric Partial
 Date Container Columns - All Releases 4-11

SQL Scenario 7: Legacy Alphanumeric Partial Date
 Container Columns - EDA 3.3.2 (MVS) & 4.x (All Platforms)..... 4-13

SQL Scenario 8: Legacy Alphanumeric Partial
 Date Container Columns - EDA 3.3.1 & Below 4-14

TABLE Scenario 1: Full Date Container Columns
 (Numeric & Alphanumeric) - All EDA Releases 4-15

TABLE Scenario 2: Implicit Century Partial
 Date Container Columns - All EDA Releases 4-16

TABLE Scenario 3: Legacy Partial Date Container
 Columns (Numeric & Alphanumeric) - All EDA Releases..... 4-17

Data Processing Considerations: 4-19

Other Considerations:..... 4-19

Level 2 - EDA Server Considerations 4-20

 Stored Procedure Considerations 4-20

 Locating Server Procedures 4-20

 MVS..... 4-20

 VM..... 4-20

 UNIX 4-21

 NT 4-21

 Reviewing Stored Procedures 4-21

 Metadata Considerations 4-24

 Locating Server Metadata 4-24

 MVS..... 4-24

 VM..... 4-24

 UNIX 4-25

 NT 4-25

 Reviewing EDA Metadata Descriptions for Dates..... 4-25

Level 3 - Database Considerations 4-26

 Determining Whether to Remediate..... 4-26

 Should you reload the database to store the century explicitly? 4-26

 Locating Server Data..... 4-27

 MVS..... 4-28

 VM..... 4-28

 UNIX 4-28

 NT 4-28

How to Remediate..... 4-28

 Sliding Date Windows – DEFCENT and YRTHRESH..... 4-29

 Setting DEFCENT and YRTHRESH..... 4-29

 1. In the EDA Server Global or User Profile 4-30

 2. At the File Level in an EDA Master File Description..... 4-30

 3. At the Field Level in an EDA Master File Description..... 4-31

 4. In a DEFINE Field in the EDA Master File Description 4-31

Table of Contents

5. On the EDA Server, Within a Stored Procedure	4-32
Order of Execution, By DEFINE Level	4-32
When Will DEFCENT and YRTHRESH Be Used?	4-33
SQL Scenario 5:	4-33
SQL Scenario 6 and 7:	4-34
SQL Scenario 8:	4-36
TABLE Scenario 1:	4-38
TABLE Scenario 2:	4-39
TABLE Scenario 3:	4-40
Creating a Century Aware Extract File for Files With Dates That Do Not Include the Century	4-43
Platform Specific Considerations for EDA	4-47
VMS EDA Considerations	4-47
VMS Operating System Considerations	4-47
Testing in EDA	4-47
Staffing	4-48
Test Environments	4-48
Test Dates	4-49
Test Dates for Applications	4-49
Test Dates for Hardware:	4-50
Test Dates for Systems Software:	4-51
Summary of Testing Issues	4-52
Testing Checklist	4-52
Source Control and Management	4-52
Unit Testing Tasks	4-52
Business Cycle Testing	4-53
Test Coverage	4-53
Other Considerations	4-53
Summary of New Date Features	4-54
Release 3.3.1	4-54
Date Handling for the Year 2000 In EDA	4-54
Release 4.2.1	4-54
Frequently Asked Questions	4-56
1. What releases of the EDA/API, EDA/Client or EDA/ODBC Extender software are Year 2000 capable?	4-56
2. Will expiration date zaps be provided to extend the expiration of older EDA Server releases like the UNIX EDA 1.x and MVS EDA 2.x servers in the Year 2000?	4-56
3. What release of the EDA Server is considered to be the Year 2000 capable release?	4-56
4. What is the first release of the EDA Server that supports the DEFCENT and YRTHRESH parameters?	4-56
5. To what field formats can I apply DEFCENT and YRTHRESH?	4-57
6. What happens if you code DEFCENT and YRTHRESH, but the field already contains the century?	4-57
7. Is there an expiration date for an MVS Release 3.3.1 server?	4-57

Table of Contents

8. How can I get the current system date returned, including the century,
when &DATE only returns a two-digit year along with the month and day? 4-57

9. What is the base date EDA uses for Smart Date format fields? 4-57

5 Enterprise Copy Server 5-1

Year 2000 Capable Releases and Enhancements 5-1

Non-Year 2000 Capable Releases 5-1

Year 2000 Testing for Copy Manager Applications 5-2

6 FOCUS for UNIX 6-1

Year 2000 Capable Releases 6-1

Commands 6-1

Frequently Asked Questions 6-1

1. Are you aware of any Year 2000 specific bugs
affecting Release 6.8.5 of FOCUS for UNIX? 6-1

2. Are you aware of any &variable issues affecting
Release 6.8.5 of FOCUS for UNIX? 6-1

3. Are there any new Year 2000 related subroutines? 6-1

4. Are you aware of any platform-specific issues
affecting Release 6.8.5 of FOCUS for UNIX? 6-2

5. Where can I find more Year 2000 related documentation? 6-2

Non-Year 2000 Capable Releases 6-2

Commands 6-2

Frequently Asked Questions 6-2

1. Will users be able to pass Year 2000 dates to the DBMS? 6-2

2. Will the data access engine still function? 6-2

3. Will subroutines that read the system clock, such as TODAY
and HHMMSS, display the correct, current, Year 2000 date? 6-3

4. Are there any &variable issues with Release 6.5.5? 6-3

5. Are there any new Year 2000 subroutines? 6-3

6. Will users be able to write trace or log files if tracing is necessary? 6-3

7. Are you aware of any issues with 3rd party communications that might be affected? 6-3

8. Are you aware of any issues with front-end application tools that might be affected? 6-3

9. Are you aware of any issues where 3rd party hardware or
operating system software might be affected by FOCUS for UNIX? 6-4

Testing and Other Issues 6-4

Third Party Issues 6-4

System Date 6-4

Environmental Issues 6-4

For Further Information 6-5

7 WebFOCUS Server 7-1

WebFOCUS Year 2000 Capable Releases and Enhancements 7-1

8 FOCUS on OpenVMS 8-1

Year 2000 Compatible Software and Hardware 8-1

IBI Year 2000 Compatible Software Available for VMS 8-1

Year 2000 Compatible Hardware Platforms Available for VMS 8-1

Table of Contents

Year 2000 Commands and Features for OpenVMS	8-1
Year 2000 Remediation.....	8-2
Year 2000 Bugs.....	8-2
Year 2000 Subroutine Issues.....	8-2
Sample Code Techniques for Year 2000 Remediation	8-2
Environmental Issues and Year 2000 Compliance.....	8-3
Year 2000 Testing	8-3
Year 2000 Testing by IBI.....	8-3
Year 2000 Testing by Customers	8-3
Leap Year Testing by Customers	8-4
Additional Year 2000 Test Dates	8-4
Non-Year-2000 Capable Releases.....	8-5
Release 6.8.6.....	8-5
Release 6.8.5 and below.....	8-6
Third Party Issues.....	8-6
Third Party Databases	8-6
VAX.....	8-6
Alpha.....	8-7
Third-Party Compatibility Issues	8-8
For Further Year 2000 Information.....	8-8
9 FOCUS Desktop	9-1
Determining If You Have a Year 2000 Problem.....	9-1
Remediation	9-1
Remediation In FOCUS For Windows (FFW).....	9-1
Remediation In FFW 6.4x.....	9-1
YY.....	9-2
CY.....	9-3
Amper Variables	9-3
Converting Your Application's Date Variables	9-5
A Note About Smart Dates	9-5
Converting Full Month and Quarter Containers Back to Full Dates	9-6
Converting Databases.....	9-6
Converting the EMPDATA Database	9-6
Converting the EMPLOYEE Database	9-7
Remediation in FOCUS Desktop (FFW7.x)	9-8
Testing.....	9-11
10 Maintain.....	10-1
Year 2000 Dates and Maintain.....	10-1
Getting Today's Date	10-1
Converting Data With Maintain	10-2
Scenario 1:	10-2
Scenario 2:	10-4
11 SmartMode/SiteAnalyzer	11-1
Year 2000 Capable Releases and Enhancements	11-1

Table of Contents

Non-Year 2000 Capable Releases.....	11-1
12 Year 2000-Related Known Problems.....	12-1

1 Introduction

Getting Ready for the New Millennium, Year 2000 Remediation for Information Builders Products, is intended for individuals involved in the installation, maintenance, use, and support of IBI products. After reviewing this document, IBI employees and customers alike will have a better understanding of how Year 2000 issues will affect current and future applications. This document will help you identify issues applicable to the Year 2000 problem, as well as provide preventive measures that can be taken before such issues affect your systems and business. Although there is no guarantee that your applications will be fully Year 2000 compliant simply by following a few steps, this manual will prove to be extremely useful to your remediation efforts. In some cases, no changes will be necessary to make an application Year 2000 compliant. In others, extensive effort will be required.

Note: Before you begin any Year 2000 related work involving Information Builders products, please see Year 2000 Compliance Issues Statement on page 1-6. The most recent version can be found on our web site, at: http://www.ibi.com/solutions/y2k/y2k_index.html.

IBI provides further material regarding the impact that the new millennium will have on systems and applications. This document incorporates insight from our own technical experts on Year 2000 issues taken from previous documentation. Several references that should be mentioned are: the September/October 1998 issue of the *Information Builders Systems Journal*, Volume 11, Numbers 4 and 5 (dedicated to the Year 2000 problem), the Information Builders Professional Services Course 860, Year 2000 Remediation, and various Information Builders product documentation (for example, the *FOCUS for IBM Mainframe User's Manual*).

The *Systems Journal* issue mentioned above is devoted to Year 2000 concerns and contains more than 25 articles on the subject. Course 860 is useful not only because it permits interaction on the subject with a knowledgeable instructor, but real life experiences are used to convey concepts and techniques in even more detail. See our web site, www.ibi.com, for information on how to subscribe to the *Systems Journal* or to register for Course 860.

Also noteworthy are two IBI products, WALDO and SiteAnalyzer. Both of these products can help you in the process of remediating your applications. References are made to these products in this document, but you may obtain further information in each product's manual, or by calling your local Information Builders sales office.

The structure of this document is designed to help you determine what you need to do based on the nature of your application, what IBI product you are using, your current release and your environment. Specific sections within this document address frequently asked questions, generic FOCUS issues, FOCUS for the IBM mainframe, EDA, platform specific issues, and Year 2000 issues relating to other IBI products. The document will be updated periodically throughout 1999. It is also available on the web in the technical support section of www.ibi.com.

Introduction

It probably will not be necessary for you to read this entire document to understand how to evaluate and repair an application. At minimum, you should read the introduction and the chapters on the Information Builders product that was used to develop that application. Scanning through the Table of Contents will help you pinpoint the information you need within each chapter, as will the index. The pages of this document are numbered by section and page number within each section. For example, page 3-14 is the fourteenth page of section three.

If after reading this book you have any further questions about the Year 2000, and how it may affect you and your Information Builders products, please contact Information Builders Customer Support Services at 800-736-6130, or via the internet at www.ibi.com.

Feedback

The Support Technology Group welcomes your comments and suggestions regarding Information Builders products and the Year 2000 problem. Information Builders can neither acknowledge nor remunerate contributors.

Updates

This publication will be updated throughout 1999 as new information becomes available. To see the most recent version, as well as a log of changes, visit the Information Builders Technical Support Site at: http://techsupport.ibi.com/index_out.asp.

Acknowledgements

This book could not have been assembled without the invaluable help of the IBM Programming and Development, EDA, Desktop Solutions and Open Systems divisions; the Information Support Services, Technical Writing, and Online Systems and Support groups of Documentation Services; the Design, Expediting and Administrative Services groups of Publishing Services and the Customer Support Services staff of Corporate Technical Support Services. Crucial feedback was provided by all of the above as well as several Information Builders customers.

Year 2000 Basics

The terms used to describe Year 2000 issues and solutions can be very confusing. Just what does Year 2000 capable mean? What is Year 2000 compliance? What is remediation? What do these terms mean in the context of Information Builders products and the applications developed with these products? When using the terms capable, fully capable, compliant or remediation, we need to understand whether we are referring to an application or to a given product's capabilities. This section addresses these questions in preparation for the more detailed sections on problem identification and remediation that follow.

Applications

Applications consist of programs, metadata (Master File Descriptions), and actual data. When referring to applications, the term compliant is typically used, but the terms capable and compliant really mean the same thing. The reason for this is that an application either handles Year 2000 issues properly, or it does not. If an application is Year 2000 compliant, it is able to consistently and accurately read, store, calculate and display valid date values before and after December 31, 1999.

In other words, an application is Year 2000 compliant when:

- It operates correctly before, during and after Year 2000.
- Date representations contain four-digit years. This is known as explicit compliance. The century is accepted as part of the date and stored correctly.
- Date representations that contain two-digit years are unambiguous in their context. In other words, if a date is input without century information, a set of rules assigns that date an accurate century value so it can be processed correctly. This is known as implicit compliance.

If an application is not Year 2000 compliant, your options are to:

- Retire the application. Simply stop using it.
- Replace it. Install a new, Year 2000 compliant application in place of the old one.
- Repair or remediate it. Fix the application you have.

If you developed an application using an Information Builders product, your ability to repair or remediate it is dependent, in part, on using a Year 2000 capable release of that product. The most recent releases of Information Builders software incorporate tools that will enable you to choose from the following repair options:

- Date expansion
- Date conversion
- Sliding window techniques

Introduction

- Application changes

These approaches are examined in detail in the following sections. Depending on the release of the IBI product you are using, the options open to you and the flexibility you have will vary. Whichever of the options you choose, you need to understand that making an application Year 2000 compliant may require changes to more than the programs. You may have to change the data and the metadata as well.

Products

When discussing products, the terms capable, fully capable, compliant and remediation reflect the degree to which the products can help you develop Year 2000 compliant applications or repair existing applications to become Year 2000 compliant. Unfortunately, these terms are often loosely defined, and may vary from product to product. Generally speaking, you should at least be using a release that is Year 2000 capable. However, it is really the nature of a given application that will dictate the product features you need.

Although certain applications may not need all the Year 2000 capabilities of the latest release of a product, you may want some of them because they make development or remediation easier. On the other hand, even if you have the latest release of a product, and it has all the capabilities you could want, you could wind up developing an application that is not Year 2000 compliant. Therefore, instead of worrying whether a release is capable, compliant, or otherwise, you should concentrate on knowing what features you need or want for your specific situation, what release of the product provides these features, and how to use these features properly.

The good news is that all recent releases of IBI products, and many older releases as well, are Year 2000 capable. Even some older releases that IBI is not officially calling capable can be used to develop Year 2000 compliant applications. When using four-digit years, otherwise known as full date containers, all production versions of FOCUS and EDA feature explicit date compliance, as well as what is known as general integrity. This means that using 21st century dates, or changing the system date to on or after 01/01/2000 will not cause the product to operate abnormally. In this way, Year 2000 compliant applications can be developed by using four-digit years in these releases. In fact, FOCUS has always been Year 2000 capable, given that it has always been able to accept four-digit year specifications such as A8YYMD or I8YYMD. However, A8 and I8 fields lack automatic date validation, and subroutines must be used for date math.

Smart Dates, available since FOCUS for IBM Mainframe Release 5.5, provide a variety of display formats, explicit compliance with four-digit years, automatic validation, commands that perform date math and correct sorting regardless of date format. However, with two-digit years, Smart Dates automatically assume that the missing century value is 19.

Remediation

Remediation is the process of modifying an application to become Year 2000 compliant. The data is changed to contain unambiguous date values, code is changed to correctly display, sort, select and store unambiguous dates, and the application is upgraded to run on a Year 2000 capable release level of IBI software.

There are two basic approaches to Year 2000 remediation: application oriented and data oriented. An application-oriented Year 2000 remediation changes the data processing procedures in the application so they can accurately process ambiguous dates. An example would be the DEFCENT and YRTHRESH functions in FOCUS. Combined, these functions automatically assign century values to two-digit dates. A data-oriented remediation changes the data data itself to make the century value explicit. An example would be expanding the year field of every date in the database to accommodate a four-digit year. Often changes to the database will require changes to the data processing application as well. Both application- and data-oriented approaches are discussed in this document.

In some cases, you might hear that a particular release of a product is being called a remediation release, such as FOCUS for IBM Mainframes Release 7.0.8R. A remediation release provides features that make remediation much easier, such as the new REBUILD facility and new DATE functions for Smart Dates (for example, &YYMD, in 7.0.8R). Note that you may still be able to remediate an application using a release that has not been so labeled - it just won't be as easy. Neither EDA nor the Desktop Division currently have a remediation release. However, remediation is still possible on both of these platforms.

Summary

The Year 2000 Compliance Issues Statement lists Information Builders products and how they can be defined in terms of integrity criteria such as general, date, implicit and explicit integrity. The statement that was available as of this printing follows, but the most current form can be found at: www.ibi.com/solutions/y2k/y2k.html . Overall, it is better to place more emphasis on considering your application, your data, and your environment, than it is to concentrate on whether a given release of a product is called the capable, compliant or remediation release. Once you have a clear picture of your needs, you will be better able to determine which releases of IBI products have the features you need to make your applications Year 2000 compliant.

Year 2000 Compliance Issues Statement

Revised November 23, 1998¹

Background

Since the mid 1980's Information Builders' software has provided date containers for date fields which are capable of carrying century information. This feature is embodied in date formats represented as MDY, YYMD, etc. which is documented in the relevant Users Manuals for each product release. These formats are 'full date containers'. A value for a century in a 'full date container' is described as 'explicitly supplied' when an application is populated with four-digit years. All manipulations with these 'full date containers', such as comparisons, sorting, indexing, date arithmetic, and display will produce correct results in a single century or across multiple centuries. Applications based upon full date containers where the century was explicitly supplied have the ability to operate correctly into the next century. Applications where either the full date facility was used, but did not explicitly supply the century to the date container, e.g. only two digits of the year were supplied, or the full date facility was not used, require features to also allow these applications to be able to operate in the next century. The main features needed are the 'century threshold window', and the 'default century value'. The integrity capabilities (as defined in Table 1) of the various releases that provide full date containers is shown in Table 2. Applications which do not use full date containers, but use older date methods, may still be made to operate correctly into the next century with the enhancements in the newer releases. Attention has been given to making remediation easier for applications using old dates and this effort is ongoing. Table 3 summarizes the releases which are being given this attention.

Criteria Framework

In order to clarify the differences in these releases we have used as a framework the integrity criteria for defining Year 2000 century issues which are summarized in a paper from GTE Government Systems Corporation, entitled *Millennium 2000 Program Document PA96014 Revised 1996-06-19*.

Table 1 summarizes the four overall criteria.

Table 1: Year 2000 Date Integrity Criteria

CRITERION	DESCRIPTION	
General integrity	No value for current date will cause interruptions in normal operation.	e.g. high risk date like Dec 31, 1999 rolls to Jan 1, 2000.
Date integrity	All manipulations, (sorting, indexing, comparing, calculating, etc.) of calendar related data (dates, durations, days of week, etc.) will produce desired results for all valid date values within the application domain.	e.g. day of week computes across centuries.
Explicit century	Date elements in interfaces and data storage permit specifying century to eliminate date ambiguity.	e.g. if the century is explicitly supplied product uses it.
Implicit century	For any date element represented without century, the correct century is unambiguous for all manipulations involving that element.	e.g. if the century is not supplied it can be inferred from the 'century threshold window' or 'default century'.

Table 2: Integrity Capability for Applications Using Full Date Containers

FOCUS

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
BULL	GCOS	6.7	Y	Y	Y	N
DIGITAL	UNIX	6.8.6	Y	Y	Y	Y
	VMS	6.9.3	Y	Y	Y	Y
		6.8	Y	Y	Y	N
HP	HP-UX	6.8.6	Y	Y	Y	Y
		6.8.5	Y	Y	Y	Y
	MPE	6.5.2	Y	Y	Y	N
IBM	AIX	6.8.6	Y	Y	Y	Y
	MVS/ CMS	7.0.8r	Y	Y	Y	Y
		6.5-6.8	Y	Y	Y	N
	OS400	6.8.6	Y	Y	Y	Y
5.94-6.8.5		Y	Y	Y	N	
MS	NT4.0	7-7.4	Y	Y	Y	Y
	NT3.51	6.4	Y	Y	Y	N
	Windows	6.4	Y	Y	Y	N
	Win 95	7-7.4	Y	Y	Y	Y
NCR	UNIX SVR4	6.8.6	Y	Y	Y	Y
SUN	Solaris	6.8.6	Y	Y	Y	Y
		6.8.5	Y	Y	Y	Y
TANDEM	Guardian	6.7.1	Y	Y	Y	N
WANG	VS	6.5	Y	Y	Y	N
Other UNIX	All	6.5.5	Y	Y	Y	N

Year 2000 Compliance Issues Statement

FOCUS/EIS

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
MS	Windows	5.1*	Y	Y	Y	N
	Win 95	5.1*	Y	Y	Y	N

FOCUS/DB-SERVER

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
MS	Win 95	6.3**	Y	Y	Y	N
	NT 3.51	6.3**	Y	Y	Y	N

EDA

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
Digital	Alpha NT	3.3.1	Y	Y	Y	Y
	UNIX	3.3.1	Y	Y	Y	Y
	VMS	3.3.1	Y	Y	Y	Y
		2.1.2	Y	Y	Y	N
HP	HP-UX	3.3.1	Y	Y	Y	Y
IBM	AIX	3.3.1	Y	Y	Y	Y
	CMS	3.3.2	Y	Y	Y	Y
	MVS	3.3.1	Y	Y	Y	Y
	OS400	3.2.2	Y	Y	Y	Y
MS	NT	3.3.1	Y	Y	Y	Y
NCR	UNIX SVR4	3.3.1	Y	Y	Y	Y
SUN	Solaris	3.3.1	Y	Y	Y	Y
All other	All	1.x-3.2.5	Y	Y	Y	N

DATA MIGRATOR

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
All	All	3.0	Y	Y	Y	Y
		2.3	Y	Y	Y	Y

CACTUS

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
All	All	3.3.2	Y	Y	Y	Y

SMARTMODE

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
All	All	3.3.2	Y	Y	Y	Y

SITE ANALYZER

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
All	All	3.3.2	Y	Y	Y	Y

SMARTMART

PLATFORM	OPSYS	RELEASE	GENERAL INTEGRITY	DATE INTEGRITY	EXPLICIT CENTURY	IMPLICIT CENTURY
All	All	3.3.2	Y	Y	Y	Y

* Only with FOCUS for Windows 6.4 in back-end.

** Only with FOCUS for Windows 6.4 as client.

Other Products

Most other Information Builders' products, not on Table 2, have the same numerical release level as the EDA level they operate with. For example, WebFOCUS 3.3.2 is equivalent to EDA 3.3.2 for Year 2000 date capabilities. For information on any Information Builders' product without this designation or not in Table 2 or 3, contact David Boyle, Year 2000 Contract Compliance Manager, at 212-736-4433.

Other Date Representation Methods

Before full date containers were available dates could appear in applications as constructions of other data types. For example, the FORMAT of A6MDY is an alphanumeric string data type where the suffix MDY just tells the system how to display the data in the field. e.g. 031595 prints as 03/15/95. These constructed data types may also have explicit centuries like I8YYMD or A8MDYY. In order to perform calculations on these data types a library of functions were provided. For example, in order to add 30 days to an old date, in an I6YMD format the AYMD function would be used, i.e. AYMD (indate,30). In order to reduce the effort in remediating these older applications a series of enhancements have been made to the latest product releases. Two enhancements are key to this effort. The first is the upgrade of all of the old date functions to compute correctly with explicit or implicit dates across centuries. The second is the extension of the 'century threshold window', and the 'default century value' to the old date formats for the purpose of allowing the date library functions to imply the centuries. Because there are other ways users can manipulate old dates the ability of an application to run in the next century with old dates depends entirely upon how the application is written. However, it is now possible to completely remediate an application in old dates. Table 3 shows which releases have the Implicit Century feature and the revised date library functions.

Table 3: Remediation Features for Applications Not Using Full Date Containers

FOCUS

PLATFORM	OPSYS	RELEASE	EXPLICIT CENTURY	IMPLICIT CENTURY	UPDATED LIBRARY
Digital	UNIX	6.8.6	Y	Y	Y
	VMS	6.9.3	Y	Y	Y
HP	HP-UX	6.8.5	Y	Y	Y
IBM	AIX	6.8.5	Y	Y	Y
	MVS/CMS	7.0.8	Y	Y	Y
	OS400	6.8.6	Y	Y	Y
MS	Win 95	7	Y	Y	Y
NCR	UNIX	6.8.6	Y	Y	Y

Introduction

Sun	Solaris	6.8.5	Y	Y	Y
-----	---------	-------	---	---	---

EDA

PLATFORM	OPSYS	RELEASE	EXPLICIT CENTURY	IMPLICIT CENTURY	UPDATED LIBRARY
All	All	4.2.x	Y	Y	Y

(Contact local IBI office for information on release availability)

Application Remediation

Users should upgrade to versions of FOCUS and EDA which provide the levels of capability they require to make their applications Year 2000 compliant. In cases where there is no upgrade available, but full date containers are available in the installed release, then the application should be converted to use these containers, and implicit century replaced with explicit century in all situations where dates are used.

Updating to even the most capable release does not guarantee that date handling will be correct for a specific application. In all applications it is necessary to review date sensitive areas. Nothing else can correct coding patterns like... COMPUTE MYDATE/YY= 1900 + OLDDATEI6YMD/10000 ; Tables 2 and 3 refer to Information Builders' supplied components. Third party products and user written routines which interface with these components may not supply satisfactory compliance, and all customers are urged to institute comprehensive Year 2000 application remediation processes for their FOCUS and EDA applications .

NOTICE

This material represents the most current information on IBI's Year 2000 Integrity capabilities, and is subject to change without notice. It is provided to assist customers in evaluating Year 2000 issues with IBI products, and refers only to IBI products. Other products in your environment (e.g. hardware, software, firmware, etc.) which are used in combination with IBI's licensed products must properly exchange data with our products. In order for IBI products to perform at the designated Year 2000 compliance level the customer must take advantage of any Year 2000 capabilities inherent in IBI software in any affected application. IBI's obligations and your responsibilities regarding IBI's products are governed solely by the agreements under which they are licensed. The information contained herein does not constitute an extension of any warranty for Information Builders software products nor does it guarantee that any software is error-free.

¹ In accordance with the recently enacted "Year 2000 Information and Readiness Disclosure Act", Information Builders designates (http://www.ibi.com/solutions/y2k/y2k_index.html) as their official statement of readiness.

2 Remediation in FOCUS

Remediation is the process of modifying an application to make it capable of handling date values beyond 12/31/99. In a fully remediated application, the data is changed to contain unambiguous date values, the code is changed to correctly display, sort, select and store unambiguous dates, and FOCUS and the operating system it relies upon is upgraded to a Year 2000 capable release level.

What is the most direct way to become Year 2000 compliant? Change all the conventional dates to Smart Dates. The year threshold and default century are then automatically supplied, and the CHGDAT, JULDAT, GREGDT, AYMD, AYM, YM, DAxxx, DTxxx, DOWK, YMD, MDY and DMY subroutines and functions are no longer required. Sorting, screening and calculations will work, and dates are automatically validated. However, since there is not much time left between now and 2000, you might consider changing only the dates that see significant activity. That is, if a date is never used or is for display purposes only, it can be among the last of the dates changed, if it is ever changed at all. If you choose this approach, you may find our SiteAnalyzer tool very useful, as it can help identify those dates with significant activity.

Determining If You Have a Year 2000 Problem

The first step in determining if an application has Year 2000-related problems is to identify the date fields used in the application. Once identified, you'll have to do some analysis to determine if these date fields will return the correct information once we start the new millennium.

To help you scan for date fields, Information Builders offers two products: Site Analyzer and WALDO. For further information on these products, please see the section "Tools Available". In general, we want to look for date fields in the FOCEXECs and Master files. Within the FOCEXEC, date fields may be used in, but are not limited to, the following areas:

- Subroutines
- Functions
- DEFINEs
- COMPUTE s
- Headings
- Sorting
- JOINs
- Amper variables
- Dialogue Manager, using constants (i.e. -SET field='19981101';)

Some other areas that can contain date fields are:

Remediation in FOCUS

- HLI programs
- Data imported from external systems
- Data exported to other systems
- “Client” FOCEXECs that may contain references to –REMOTE or REMOTE execution statements

Once the date fields have been identified and you know they appear in the FOCEXECs, Master files and other code used in the application, you must consider the following major functional areas for possible changes:

- Data Display
- Data Entry
- Data Processing
- Data Storage

We will now look at each of these areas in more detail.

Data Display

There are two factors to consider when displaying date data:

- If your application presents date data with a two-digit year field, can users consistently and correctly infer the century for each date that might be displayed? If not, this field needs to be expanded to display the four-digit year.
- If the date field is expanded to display the full four-digit year, it will affect the width of the report. This may throw data columns out of alignment and even cause the report to exceed the boundary provided on the screen.

Data Entry

There are four aspects of data entry to consider:

- Take a look to see whether the data entry screen asks users to enter years with two digits or four. If users are allowed to enter a full, four-digit year, no changes are required. However, if the input screen only allows users to enter a two-digit year, you will need to consider whether the missing century data could cause confusion. If the answer is yes, then the input screen needs to be changed.
- As with Data Display, above, expanding the input field to accept a four-digit year may also increase the width of the input screen. Is there enough room on the screen to accommodate these changes? One solution might be to keep the two-digit input field and use a sliding date window to determine the correct century for each date.

Determining If You Have a Year 2000 Problem

- If you are using FIXFORM to input the data, the same considerations must be taken. Should the input date feature a two-digit year or four-digit year? If this data is being entered into a FOCUS file, and is it using an old date format or new (Smart Date) format?
- Finally, is the data being stored with the correct century? If not, you will also need to make some changes to the way your application stores data.

Data Processing

Here are some considerations when data processing occurs:

- Are two-digit date fields used in any date subroutines or functions? If not, then you may not have a problem. However, if you are using any date subroutines or functions (and chances are good that you are), then you will probably need to upgrade to a release level that supports a sliding window or permits an eight-digit date as input to these subroutines and functions. It is likely that you will also have to change some code after you upgrade.
- If the data uses two-digit year values, is it absolutely clear which century each date belongs to during every step of data processing? If the answer is no, you will need to change your code. If the answer is yes, you should then check to see if these two-digit date fields are being used to sort or screen data. If they are, then coding changes are required.
- Consider whether the date fields are being used in Dialogue Manager via the –SET command or &variables. Will these values need to include the century? If the answer is yes, coding changes are required.
- Are calculations being done with date fields that are not Smart Dates? If so, coding changes are required.

Data Storage

Data storage is not an issue if the correct century is stored in the date field. It does become an issue when either the century is not stored because of the field format, such as I6/A6YMD, or when the century stored was implied through a Smart Date format. If the field is an integer or alpha format, what you see is what you get. The digits displayed are those that are stored. Smart Dates for FOCUS files are stored numerically, as a displacement from the base date of 12/31/1900. If you have a Smart Date format of YMD, FOCUS automatically supplies the century as 1900. If 1900 is not the correct century, the data needs to be changed. When the century is not stored with the data, you can use FOCUS's sliding window feature, DEFCENT and YRTHRESH, to keep track of the century. Keep in mind that using sliding windows will not change existing data, but will only clarify it during data processing.

Remediation in FOCUS

As you can see, these four areas are closely tied together. A change in one area will likely affect one or more of the other areas. For example, if you change your application so that the screen allows for input of four-digit years, you would probably need to make changes to the Master file that describes the field. You may then have to change the applications that display this field so that they will also display the century. If there are any references made to this field in any other FOCEXECs, you may need to change code in those applications, as well. This is what makes any type of remediation difficult. Hopefully, this document will ease the process of making these changes by increasing your understanding of how the Year 2000 related features of FOCUS work internally, and the impact these changes will have on your application and data.

Smart Dates in FOCUS files

No matter which options you chose to make your application Year 2000 compliant, an understanding of FOCUS Smart Dates is essential. Smart Dates should be strongly considered for any FOCUS remediation. They can be used in arithmetic-type operations, keeping track of dates, sorting data properly, performing automatic validation, and displaying date data in a variety of formats.

A Smart Date is a field that has one of the following formats:

YYMD	YMD	YQ	YM	JUL
DMYY	DMY	YYQ	MY	YYJUL (available in 7.0.8R on the Mainframe)
MDYY	MDY	QY	YYM	
		QYY	MYM	
		Q		

Along with these field formats, one or more of these edit options may appear: T, t, TR, tr, W, and w. For more detailed information about these editing options, see *Describing Non-FOCUS Data Structures* in the *FOCUS Users Manual*.

Any of these formats is correct for a FOCUS Smart Date, as long as the format contains the month, day and year, as well as it internally represents the date as a numerical displacement from the base date of December 31, 1900. If the format uses only a month and a year, then the base date is January, 1901. Through this format, the century is always implied. If the format for the field is a Smart Date with a four-digit year (for example, YYMD or MDYY, or DMYY), then the century is always explicitly present in the field.

Determining If You Have a Year 2000 Problem

Please note that the formats Y and YY are not Smart Dates because their internal representations are not offsets from the base date 12/31/1900. Internally, dates in Y and YY formats are stored as two and four-digit integers, respectively. A field with a Y format and a value of "99" is only "99" internally. It has no implicit or explicit century value, so it will neither sort or do the selection using "IF/WHERE" properly. On the other hand, a field with a YY format and a value of "2000" has a value of "2000" internally and will sort properly. The DEFCENT and YRTHRESH commands can be used to DEFINE a field from a Y to YY format.

Here is an example of where the output of the comparison between two Y fields is not as expected:

```
DEFINE FILE CAR
  YYMDFLD/YYMD = 20000101;
  YFLD/Y       = YYMDFLD;
  YYDATE/YY    = 1987;
  YDATE /Y     = YYDATE;
  XTEST/A3     =IF YDATE LT YFLD THEN 'LT' ELSE IF YDATE EQ YFLD THEN 'EQ'
  ELSE 'GT';
END
SET DATEDISPLAY=ON
TABLE FILE CAR
PRINT YYMDFLD YFLD YYDATE YDATE XTEST
BY CAR NOPRINT
IF RECORDLIMIT EQ 1
END
```

Here is the output:

YYMDFLD	YFLD	YYDATE	YDATE	XTEST
-----	----	-----	-----	-----
2000/01/01	00	1987	87	GT

You might think this output is wrong since the Year 1987 is less than the Year 2000. However, because we converted the fields to Y formats, the value stored is only a two-digit integer. FOCUS is comparing the values '87' to '00'. The field XTEST is correctly evaluating the values since '87' is in fact greater than '00'.

If you change the request to use the 'YY' field in the test instead, you will get the correct output.

```
DEFINE FILE CAR      YYFLD      YYDATE      YDATE      YDATE
YYMDFLD/YYMD =      -----      -----      -----      -----
20000101;
YYFLD/YY      =
YYMDFLD;
YYDATE/YY    =
1987;
YDATE /Y     = YYDATE;
XTEST/A3=IF
YYDATE LT YYFLD
THEN 'LT' ELSE
```

Remediation in FOCUS

```
IF YYDATE EQ
YYFLD THEN 'EQ'
      ELSE
'GT' ;
END
YYMDFLD
-----
2000/01/01      2000      1987      87      LT
```

Smart Dates With the Century

If you are using Smart Dates in FOCUS files with the century portion explicitly supplied, and the data entry form already provides for the century, then the width of the report will remain unchanged and there is no need for coding changes. However, if the field was redefined so that the user only has to enter the last two-digits of the year, then a default century value of '19' is probably stored in the database. See *Smart Dates Without the Century* on page 2-8 to understand how these date fields work and to see the implications this might have on your application. Is the century stored correct for these fields? To view the internal storage value of the date field, print it with a format of I9. For example, say the Master File Description includes a field called HIRE_DATE, with a format of MDYY. To display the internal storage value, consider the following code:

```
TABLE FILE fn
PRINT HIRE_DATE HIRE_DATE/I9
AS 'ELAPSED DAYS'
END
```

Sample output will look like this:

HIRE_DATE	ELAPSED DAYS
06/02/1980	29008

The second column shows the internal storage value of 29008. This is the number of days elapsed since 12/31/1900.

An example of how FOCUS is able to differentiate between two dates stored within different centuries is shown below:

Determining If You Have a Year 2000 Problem

```
DEFINE FILE CAR
DT1/YMD WITH COUNTRY = 'JANUARY 01 1900';
DT2/YMD WITH COUNTRY = 'JANUARY 01 2000';
DT3/YMD = DT1;
DT4/YMD = DT2;
DISP1/I6 = DT1;
DISP2/I6 = DT2;
END
TABLE FILE CAR
PRINT DT3 AS 'YYMD,for 1900'
      DT1 AS 'YMD,for 1900'
      DISP1 AS 'DISPLACEMENT,for 1900' OVER
      DT4 AS 'YYMD,for 2000'
      DT2 AS 'YMD,for 2000'
      DISP2 AS 'DISPLACEMENT,for 2000'
BY COUNTRY NOPRINT
IF RECORDLIMIT EQ 1
END
```

Output:

YYMD FOR 1900	YMD FOR 1900	DISPLACEMENT FOR 1900	YYMD FOR 2000	YMD FOR 2000	DISPLACEMENT FOR 2000
-----	-----	-----	-----	-----	-----
1900/01/01	00/01/01	-364	2000/01/01	00/01/01	36160

Although the YMD output appears the same on the report regardless of whether the century is 1900 or 2000 we can differentiate internally between 2000/01/01 and 1900/01/01 by using Smart Dates. The displacement automatically implies the century. In addition, arithmetic with these dates will utilize the correct century value. Remember that the year 1900 was not a leap year, but the year 2000 will be a leap year.

```
DT1 =00/01/01
DT2 =00/01/01
DT1 + 365 = 1901/01/01          DT1 - 100 = 1899/09/23
DT2 + 365 = 2000/12/31        DT2 - 100 = 1999/09/23
DIFFERENCE BETWEEN DT1 and DT2= 36524
```

Again, because of the way Smart Dates are stored, sorting and selection tests work without a problem. For more information on how this works, see the sections *Sorting*, and *Selection Tests*, Chapter 3, *Remediation in FOCUS for IBM Mainframe*, page 20.

Smart Dates Without the Century

You have seen that Smart Dates automatically contain the century value, and take it into account during arithmetic, sorting and other operations. This is true on every platform and release of FOCUS since Smart Dates were introduced in Release 5.0 of Mainframe FOCUS. If your Smart Dates have a format that includes the century, explicitly YYMD, MDYY, and DMYY, then you will not need to make changes to the smart date code to make your program capable of handling dates in the next century. However, if the century was implied then read on.

What problems will you encounter if your Smart Dates do not explicitly contain the century? Obstacles arise when you load Smart Dates with two-digit year values. Since the century value in the Smart Date is encoded through the stored date's displacement from base date, FOCUS must determine what century to encode if the stored date only has two digits. Of course, if the input contained the century, there is no problem. Notice the DEFINE above for DT1 and DT2. Although the output displays only YMD, the outcome is accurate, since the input contained the century digits. Often, however, this is not the case. You will need to determine the century digits in the following cases:

1. Data Maintenance through FOCUS where the transaction input contains only a 2-digit year.
Note: This is true with a CRTFORM containing a <field or <T.field when that field has a YMD format. This includes maintenance on not only FOCUS files, but also SQL-type tables, where the field is a SQL date field (ACTUAL = DATE).
2. DEFINE or COMPUTE of a Smart Date from a conventional date of 6 characters (not containing century).
3. An IF or WHERE test, where the literal supplied does not contain century digits.

For instructions on handling these kinds of situations, consult the November 1989 and March 1990 issues of the *Information Builders Systems Journal*. In these issues, you will find a two-part article by John Price entitled, "Using New Dates." This article provides detailed examples of converting to Smart Date formats, calculations and translations. It also describes how to use dates with Dialogue Manager, and the impact that dates have on HOLD and SAVE files, REBUILD and more.

Smart Dates in Non-FOCUS Files

The Master File Description tells FOCUS how to interpret the data in a file. For non-FOCUS files, there are two format attributes for the fields: ACTUAL and USAGE. The ACTUAL attribute describes the format of the field as it exists in the external file. The USAGE attribute describes how FOCUS will use the field for display or calculations. The ACTUAL attribute is unnecessary for FOCUS files.

For date fields, FOCUS permits the following conversions from ACTUAL format to USAGE format:

ACTUAL	ACTUAL Definition	USAGE
--------	-------------------	-------

Determining If You Have a Year 2000 Problem

DATE	Four-byte integer internal format, representing the difference between the date to be entered and the date format base date.	Date format
An	Where "n" = 1-256. Alphanumeric characters A-Z, 0-9, and the special characters in the EBCDIC display mode.	An, D, F, I, P, date format
In	Binary integers: I1 = single-byte binary integer. I2 = half-word binary integer (2 bytes). I4 = full-word binary integer (4 bytes).	I, date format
Pn	Where "n" = 1-16. Packed decimal internal format, where "n" is number of bytes, each of which contains two digits, except for the last byte which contains a digit and the sign (+ or -). For example, P6 means 11 digits plus a sign.	P, date format
Zn	Where "n" = 1-31. Zoned decimal where "n" is the number of bytes and the low order byte contains the sign.	REVIEWER: Should Z, date format appear here?

Note: For date fields that have an ACTUAL of either DATE, An, In or Pn, the USAGE attribute can be a Smart Date. This is noted by the term "date format" in the USAGE column and is documented as such in the User's Manual.

For non-FOCUS files, FOCUS can only control how the data is displayed or used internally in FOCUS. FOCUS cannot change the actual storage of the field. The table below shows how the various USAGE formats control how the value is displayed in a report, even as the stored value remains unchanged.

USAGE	ACTUAL	Value	Display
WrMtrDYY	A6YMD	851230	Monday, December 30 1985
YQ	A6YMD	991230	89 Q4
QYY	A6YMD	001230	Q4 2000
YMD	A6	001230	00/12/30
MDYY	A6YMD	991230	12/30/1999

For more information on describing data, please see *Describing Non-FOCUS Data Structures* in your Users Manual.

Tools Available:

Using FOCUS Code to Scan for Date Fields

- If you do not have access to any tools that scan for date fields, you can use FOCUS code to help you determine which FOCEXECs, MASTERS and COBOL FDs contain dates. However, be aware that date fields may not always be obvious, especially if the code has gone through several iterations of DEFINE. Here are some date formats to look for: I2, I4, I6, as well as A2, A4, and A6 (without any references to a year, month or date).
- Julian dates or date fields where an ALIAS may have been used.
- Date amper variables, i.e. for example, &YMD, &MDY, &DMY and &DATE. You should check these variables to see if the century needs to be added to the amper variable to make its value unambiguous.
- Date functions/subroutines, including YMD, MDY, DMY, AYMD, AYM, YM, CHGDAT, JULDAT, GREGDT, DAxxx, DTxxx, DOWK and TODAY.
- Components of smart date formats (M, D, Y and any combinations of these formats).
- Date constants or literals.
- Variable names containing date-related strings, such as 'YEAR', 'YR', 'DATE', 'BEGIN', 'END', 'BIRTH', etc.

Try the following procedure. Based on your operating system, create a file with the list of all pertinent MASTER files. In CMS, use LISTFILE; in MVS, use FOCMAP; in PC, use DIR piped to a file. Now, set up a loop in Dialogue Manager:

```
CMS FILEDEF INPT DISK CMS EXEC A
-RUN
-LP
-READ INPT NOCLOSE &FILL.A4. &FN.A8.
-IF &IORETURN NE 0 GOTO NOMORE;
CHECK FILE &FN HOLD
TABLEF FILE HOLD
HEADING
"DATE FIELDS IN FILE &FN "
PRINT FIELDNAME OVER ALIAS OVER USAGE
BY SEGNO NOPRINT BY SEGNAME BY FLDNO NOPRINT
WHERE USAGE CONTAINS 'Y'
END
-GOTO LP
-NOMORE EXIT
```

Tools Available:

This code will loop through all the MFDs created by the initial step and create a report of date fields. This works because a date format will always contain a 'Y'. Similarly, create a Master File Description that we'll call FEX:

```
FILE=FEX,SUFFIX=FIX
SEGNAME=SEG1,SEGTYPE=S0
FIELD=THELINE,,A80,A80,$
DEFINE LINENO/I4 WITH THELINE = LINENO + 1;
```

Now, COBOL copy members, FOCEXECs, and the like can all be allocated or FILEDEFed to the ddname, FEX. Then issue:

```
TABLEF FILE FEX
PRINT THELINE BY LINENO
IF TOTAL THELINE CONTAINS 'DATE' OR 'DTE' OR 'YEAR' OR ....
END
```

Assuming that the fieldnames for date fields contain some variation of the word DATE, this will help you estimate the scope of the problem.

Another example of this type of coding can be found in an article entitled "Y2KCHECK: A FOCUS Master File Description Scanning Utility," by Chris Bell in the September/October 1998 issue of *Information Builders Systems Journal*, Volume 11, Numbers 4 and 5, page 165.

SiteAnalyzer

SiteAnalyzer is a collection system that reports on all FOCUS queries. All applications run normally as the collection phase takes place. Then SiteAnalyzer creates reports from these collection files which help to identify references to date fields. SiteAnalyzer can help identify the following:

- Whether the application is a MODIFY or TABLE.
- Whether the date field is a new or old format.
- How the date is being used.
- Whether the field is a real field, DEFINE in the Master File Description, or a DEFINE/COMPUTE.
- Subroutines using date references.
- The user id, name of the Master File Description, and the name of the FOCEXEC.
- If the application was run online or in batch mode.

See "Solving the Year 2000 Problem: SiteAnalyzer Provides Real Help," by Jack Kissner, on page 28 of the September/October 1998 issue of the *Information Builders Systems Journal*. This article documents how SiteAnalyzer identified and determined the scope of the Year 2000 problem at an Indiana insurance firm, and how it helped them change their applications to become Year 2000 compliant.

WALDO

WALDO (Where Are Likely Date Objects) is Information Builders' Year 2000 impact analysis tool for FOCUS. Unlike other FOCUS Year 2000 code scanners, WALDO installs on the FOCUS host and runs where FOCUS code resides, minimizing disruptions and reducing the likelihood that files will be overlooked. WALDO operates in either online or batch mode, to parse, analyze and catalog all active, static, production and user-written FOCUS code. WALDO's efficient parsing engine is user-customizable and tunable, so users can easily locate custom dates with unique formats and field names in FOCUS applications.

WALDO supports enterprise-wide Year 2000 remediation efforts with its special repository, project-management facilities, and detailed reports. Its repository catalogs date fields used, as well as the Master Files, FOCEXECs and statements impacted by each date – all organized by user department and application. WALDO's comprehensive management functions can also create detailed reports for monitoring the progress of Year 2000 remediation efforts, including impacted FOCUS code, the staff responsible for fixing it, and the status of each FOCEXEC and Master File Description (MFD).

In mainframe environments, WALDO can import the names and locations of FOCUS libraries identified by Information Builders' sophisticated SiteAnalyzer system-management tool. This enables MVS or VM FOCUS users to combine WALDO's line-by-line risk assessment with SiteAnalyzer's usage-based insights to efficiently identify, prioritize and manage Year 2000 projects.

WALDO is available for Year 2000 impact assessment and remediation efforts on all mainframe and midrange FOCUS platforms, including MVS, VM, OpenVMS, and UNIX environments such as HP-UX, AIX, NCR, Digital and Solaris.

For more information on WALDO, please contact your local sales office, or call 1-800-969-INFO.

Professional Services

Information Builders' Professional Services Group helps to insure the success of our customers by providing integration, advisory and training services through a network of 85 worldwide branch offices. Our experienced consulting professionals apply comprehensive, proven techniques to the process of transforming real-world business problems into efficient information systems. Specialized consultants can assist in your Year 2000 conversions. In addition, Information Builders offers Course 860, Year 2000 Remediation, which addresses the Year 2000 problem and FOCUS remediation techniques.

For more information on Professional Services, please contact your local sales office, or call 1-800-969-INFO.

Tools Available:

Systems Journal

Information Builders Systems Journal is a bi-monthly technical publication that users of Information Builders' software and services count on for the latest tips, techniques and solutions vital to improving the performance of any business information system. The September/October 1998 issue of *Systems Journal* is dedicated to discussing the Year 2000 problem, and you will see several references made to it in this document.

For more information on *Information Builders Systems Journal*, please visit the Information Builders bookstore web site at www.ibi.com/bookstore/index.html, or call 1-800-969-INFO.

Remediation in FOCUS

3 FOCUS for IBM Mainframe

FOCUS for IBM Mainframe Release, 7.0.8R is called the “remediation release” because it includes additional new features specifically for converting older date fields with two-digit date formats to Year 2000 capable years with four-digit date fields. In this context, the two most significant features of Release 7.0.8R are:

- REBUILD, with the DATE NEW option.

A new option has been added to REBUILD for legacy date conversion. The option DATE NEW converts legacy dates to Smart Dates in your FOCUS databases. The utility uses ‘update-in-place’ technology. It updates your database and creates a new Master File, yet does not change the structure or size of the database. You must backup the database before executing REBUILD with the DATE NEW option. We recommend that you run the utility against the copy and then replace the original file with the updated backup.

- The new SET parameter: ALLOWCVTERR = ON/OFF (default is OFF).

If ALLOWCVTERR=ON, it will convert invalid dates to 0 (base date) or MISSING (if MISSING = ON is set for the field). Prior to release 7.0.8R, attempting to convert from an “invalid” traditional date to a “Smart Date” would result in a FOC1130 (format conversion error) or FOC200 (invalid format message). This setting only applies to non-FOCUS files during an ACTUAL to USAGE conversion in a TABLE(F) request. This does not apply to MODIFY, COMPUTE and DEFINE.

These two new features and more are discussed in detail in the FOCUS 7.0.8R Documentation CD, DN100100.0698.

What Is Remediation?

As mentioned previously, remediation is the process of modifying an application to become Year 2000 compliant. The data is changed to contain unambiguous date values, the code is changed to correctly display, sort, select and store unambiguous dates. All releases of FOCUS for IBM Mainframe are Year 2000 capable and can be used to develop applications that are Year 2000 compliant, so long as the applications include four-digit years and, if they were developed using a release prior to 7.0.8, do not include FOCUS date subroutines or FOCUS date arithmetic. Date subroutines are Year 2000 capable in Release 7.0.8 if the Mega PTF has been applied, and are capable in all subsequent releases. The following are the basic steps for ensuring an application is Year 2000 compliant:

1. Determine if you have a Year 2000 problem by identifying the date fields and how they are used.
2. Identify the available options and determine which is most suitable.
3. Make the necessary changes.
4. Conduct numerous tests using a wide range of dates to ensure that the changes are valid.

What To Do If You Are In a Year 2000 Capable Release

FOCUS for IBM Mainframe, Release 7.0.8R and Release 7.0.8 with the unsolicited Mega PTF applied, are considered Year 2000 capable. Both releases include DEFCENT and YRTHRESH. Subroutines such as AYMD, AYM, CHGDAT, GREGDT, JULDAT, DAxxx, DTxxx and the FOCUS functions, YMD, DMY, MDY, are now able to take advantage of the DEFCENT and YRTHRESH settings. First, we will address steps two and three from What Is Remediation? on page 3-1 to identify the options available, determine which options to use, and then make the necessary changes.

In FOCUS for IBM Mainframe, your options for making an application Year 2000 compliant are:

- Sliding Window
- Date Expansion
- Date/Data Conversion
- Application Changes

Sliding Window – DEFCENT and YRTHRESH

The solution to the problem of attributing a century to a date with a two-digit year field is to use the settings DEFCENT and YRTHRESH. These two new SET parameters are used to define century values for a 100-year span of two-digit year dates. This feature is available as of Release 7.0.5 on FOCUS for IBM Mainframe.

DEFCENT and YRTHRESH can be used where date conversion takes place. For example:

- Converting old dates to Smart Dates when the first two digits are not supplied.
- Converting from ACTUAL to USAGE when the first two digits are not supplied.
- Data entry (via FIXFORM or CRTFORM) when the first two digits are not provided. This is true even if the input field is eight-digit capable and only two digits are provided for the century portion.

The syntax is

```
SET DEFCENT = nn
```

where:

19

Is the default value.

```
.SET YRTHRESH=nn
```

and

00

Is the default value, and nn can be negative.

Note: The underscored values are the default values. The settings for DEFCENT and YRTHRESH can be viewed using the ? SET command.

As of Release 7.0.8R, the SET YRTHRESH parameter is enhanced to include SET YRTHRESH = - nn, where 'nn' is subtracted from the current year and the resulting year value becomes the lower boundary for the 100-year data window. This creates a moving window that automatically increments itself each year.

Here are some examples of settings and the 100-year span each represents:

DEFCENT	YRTHRESH	100-year span	2-digit year supplied	4-digit year
19	00	1900-1999	40	1940
			90	1990
19	50	1950-2049	40	2040
			90	1990
19	-38*	1961-2060	40	1940
18	-101*	1898-1997	99	1899

*Presumes that the current year is 1999.

Per the table above, if you set YRTHRESH to a value large enough to cross a century boundary, the value of DEFCENT is recalculated. For example, if you set YRTHRESH to minus 101 (1999-101 = 1898), DEFCENT is calculated to 18 and YRTHRESH becomes 98. The 100-year span begins with a pivot year of 1898 and ends with year 1997. ? SET and ? SET ALL will then reflect the latest setting of DEFCENT.

As of Release 7.0.8, there are four places where DEFCENT and YRTHRESH can be set. They are:

- Temporary Field Level (DEFINE or COMPUTE)
- Field level
- File level
- Global level

These settings are applied in the following order:

1. Temporary field level
2. Field level
3. File level
4. Global level

FOCUS for IBM Mainframe

If DEFCENT and YRTHRESH are set on the left hand side of the “=” in the DEFINE or COMPUTE statement and the output is a Smart Date, these settings will override all other levels. If values for DEFCENT and YRTHRESH are set at the field level, they will override values set at the file level. If they are set at the file level, they will override values set at the global level. The global level applies if DEFCENT and YRTHRESH are not set anywhere else.

DEFCENT, YRTHRESH and Maintain

Maintain respects field and Master level settings of DEFCENT and YRTHRESH in all releases of FOCUS for IBM Mainframe since Maintain’s introduction in Release 7.0.6. Starting with Release 7.0.8R, Maintain respects global level settings of DEFCENT and YRTHRESH, except in computed fields. Release 7.0.9 is fully remediated and respects all settings of DEFCENT and YRTHRESH.

Temporary Field Level Example

```
DEFINE FILE ALUMNI
NEW_DOB/YYMD DFC 20 YRT 04 = DT_OF_BIRTH;
END
```

Note: DFC and YRT at the DEFINE level are only used when the output is a Smart Date. The abbreviations DFC and YRT cannot be used with the SET command.

Field Level Example

```
FILE = ALUMNI, SUFFIX = FIX, $
SEGNAME = SEG1, SEGTYPE = S1
FIELD = ALUMI_NO , ALIAS = SSN , USAGE = A9 , ACTUAL = A09,$
SEGNAME = ALUMNI, PARENT = SEG1
FIELD = NAME , ALIAS = NAME , USAGE = A20 , ACTUAL = A20,$
FIELD = DT_OF_BIRTH , ALIAS = DOB , USAGE = YMD, ACTUAL = A06,
DEFCENT = 20, YRTHRESH =04
$Year Span 2004 → 2103
FIELD = GRADUATION_DT , ALIAS = GTD , USAGE = YMD, ACTUAL = A06,$
```

File Level Example

```
FILE = ALUMNI, SUFFIX = FIX, FDEFCENT = 18, FYRTHRESH = 70
$Year Span 1870 →1969
SEGNAME = SEG1, SEGTYPE = S1
FIELD = ALUMI_NO , ALIAS = SSN , USAGE = A9 , ACTUAL = A09,$
SEGNAME = ALUMNI, PARENT = SEG1
FIELD = NAME , ALIAS = NAME , USAGE = A20 , ACTUAL = A20,$
FIELD = DT_OF_BIRTH , ALIAS = DOB , USAGE = YMD, ACTUAL = A06,
FIELD = GRADUATION_DT , ALIAS = GTD , USAGE = YMD, ACTUAL = A06,$
```

Global Level Example

```

SET DEFCENT = 19
SET YRTHRESH = 10
-* Span of years 1910 → 2009
MODIFY FILE STUDENTS
FIXFORM STUDENT_ID/9 NAME/20 DOB/6 GRADUATION_DT/A6YMD
MATCH STUDENT_ID
    ON MATCH INCLUDE
    ON NOMATCH REJECT
DATA
111111111MICHAEL ANGELO      800621010620
END

```

For Smart Dates where the century is not supplied, FOCUS will, by default, use the current century (i.e. 19) when calculating the displacement from the base date. Suppose we load a student's birth date (111979), and graduation date (060501) into a database. Since both dates have a format of MDY, we load the birth date as the equivalent of November 19, 1979, and the graduation date as the equivalent of June 6, 1901. The graduation date, of course, is incorrect. However, if we code DEFCENT and YRTHRESH at the field level prior to loading the data, both the student's birth date and graduation date would be associated with the correct century.

Note: DEFCENT and YRTHRESH will not change existing data in a file. As an example, consider a file that keeps employee records at Information Builders. The file contains a field called HIRE_DATE, with a format of YMD. Typically, as soon as a person accepts the job, personnel will enter the hire date into the database. We can then assume that in December of 1999, we will enter people for hire dates in the year 2000. However, we also know that we will not hire people born in the year 2000 for many years. In the database, we can make the following changes:

```

FIELD=BIRTH_DATE, DOB, ACTUAL=I6YMD, USAGE = YMD, DEFCENT=19, YRTHRESH=30, $
span 1930-2029

FIELD=HIRE_DATE, HDT, ACTUAL=I6YMD, USAGE = YMD, DEFCENT=19, YRTHRESH=75, $
span 1975-2074

FIELD=DEPENDENT_DOB, ACTUAL =I6YMD, USAGE = YMD, DEFCENT=19, YRTHRESH=70, $
span 1970-2069

```

Why those dates? Currently, there is no company employee who was born prior to 1930. Since it is now 1999, we can assume that we will not hire anyone above age seventy in the year 2000. Since Information Builders was founded in 1975, no hire dates can possibly precede that date. Choose these numbers carefully. If we had used a DEFCENT of 19, and a YRTRHESH of 99 for HIRE_DATE, the YRTRHESH span is 1999 to 2098. Code that selects employees hired before 1980, such as WHERE HIRE_DATE LE '800101', would translate to the equivalent of WHERE HIRE_DATE LE '2080/01/01'. Everyone would appear. Again, remember that by applying DEFCENT and YRTHRESH, we will not change current, existing information. In order to change the existing data, see Date/Data Conversion on page 3-6.

FOCUS for IBM Mainframe

Note: Prior to Release 7.0.8R, if the date value is exactly equal to the base date (for example, the date you input is 12/31/1900), you will by default see blanks in the output, as we cannot differentiate between a zero value meaning “no data supplied” and a zero value meaning zero displacement. This problem was corrected with the SET parameter, SET DATEDISPLAY = ON.

This feature is available as of Release 7.0.8R. SET DATEDISPLAY = ON, will cause a displacement value of 0 to display on the output report as 12/31/00. If SET DATEDISPLAY=OFF, however, FOCUS 7.0.8R will not be able to distinguish between a zero value meaning “no data supplied” and a zero value meaning zero displacement. Also, remember MISSING=ON is valid for dates, so if a user attempts to display a FOCUS file that has no data, the MISSING data character(s) will be displayed.

Date Expansion

The date expansion option entails converting all six-character date field formats to eight characters in order to include the century. All A6 date field formats would become A8 formats, all I6 formats would become I8 formats, and so on for D6, P6, Z6 and F6 formats. Smart Date formats, MDY, YMD and DMY, would be expanded to include another “Y” resulting in the following: MDYY, YYMD and DMY Y. When date fields are expanded, they will likely affect the width of data entry screens and reports, causing columns to overlay. Remember that if you are already using Smart Dates and need to expand the field, be sure to check that the century currently stored with the field is correct. The default century value for a Smart Date with a two-digit year field is ‘19’.

Date/Data Conversion

Should you reload the database to store the century explicitly?

Before you convert your entire database from two-digit year fields to four, consider the impact this change may have on all the applications that use this file. This may not necessarily be the best approach to making the application Year 2000 capable. Using a sliding window, DEFCENT and YRTHRESH may be enough to store and calculate the century.

There are many factors to consider before you make your decision, including:

- **Are these files FOCUS files or non-FOCUS files?**

If they are non-FOCUS files, conversion takes place between the ACTUAL and USAGE formats, and you can take advantage of the sliding window technique. FOCUS data files are read directly into FOCUS applications, so no conversion takes place at access time and the data itself must be changed if it was incorrect. However, new data that is entered after supplying DEFCENT and YRTHRESH to the field or file level will be century aware. Again, simply applying DEFCENT and YRTHRESH will not change existing data.

- **Are these date fields used in any JOINS?**

If you change a date field in one file, you will also need to make corresponding changes in the target file.

- **How are these dates stored in historical databases?**

Will you still be able to do a USE on the file since the format may not be the same any more? If you change the format in one file, the USEd file must have the date field in the same format since the same Master File description is used to read the data.

- **How does this impact the screen for user input?**

Users may find it annoying to always enter a century in instances where the century is redundant or never ambiguous. There may not be room to add the space required to enter two additional bytes of data on the screen.

- **Does this affect the width of reports or the positioning of fields on reports?**

The report may already be at maximum width, or it is possible that the layout of the report will be changed considerably by expanding the date field.

- **Did you check the logic for all the applications that go against these date fields?**

IF/WHERE tests and DEFINES/COMPUTES may no longer work.

At this point, if you have decided that you want to convert the data and it is in a traditional six-digit date format, seriously consider using Smart Dates instead of eight-digit date format fields. Smart Dates can be used in arithmetic-type operations, keeping track of dates, sorting data properly, performing automatic validation, and displaying date data in a variety of formats.

How to Reload the Database to Store the Correct Century

Release 7.0.8R includes a mechanized facility to convert traditional dates to “Smart Dates” in FOCUS files. The details of using this facility are documented in the section of the FOCUS 7.0.8R Documentation CD (DN1001050.0698), *NF557: REBUILD Enhancement - Legacy Date Conversion*. For non-FOCUS files and dates in alpha format that cannot be converted using the new REBUILD facility, you can follow the steps described below.

16YMD Non-Key Field Conversion Process

1. Copy the original Master and save it to a new name.
2. Change the old date formats to Smart Date format by applying the following:

```
FILENAME=ORIGDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
FIELDNAME=DATE, ALIAS=, FORMAT=I6YMD,$
FILENAME=NEWDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
FIELDNAME=DATE, ALIAS=, FORMAT=YMD,$
```

FOCUS for IBM Mainframe

3. Copy the original database and save it to a new name.
4. Copy the original Master to create a temporary Master, retaining the original format.

```
FILENAME=TMPDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
FIELDNAME=DATE, ALIAS=, FORMAT=I6YMD,$
```

Issue a USE command for the temporary Master:

```
USE
ORIGDATE AS TMPDATE
END
```

5. Write a MODIFY to perform the conversion:

```
MODIFY FILE TMPDATE
COMPUTE DATEYMD/YMD=;
      DATEINT/I8=;
NEXT NUM
ON NEXT COMPUTE DATEYMD=D.DATE;
ON NEXT COMPUTE DATEINT=DATEYMD;
ON NEXT COMPUTE DATE=DATEINT;
ON NEXT UPDATE DATE
ON NONEXT GOTO EXIT
DATA
END
```

6. Rebuild the index if required.
7. Test the new database using the new Master.
8. Delete the original database.
9. Delete the temporary Master.
10. Rename the new database and Master.

16YMD Key Field Conversion Process

1. Copy the original Master and save it to a new name.
2. Change the old date formats to Smart Date format by applying the following:

```
FILENAME=ORIGDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=I6YMD,$
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
FILENAME=NEWDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=YMD,$
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
```

3. Copy the original database and save it to a new name.

4. Copy the original Master to create a temporary Master, retaining the original format. Change the SEGTYPE=S1 to S0.

```
FILENAME=TMPDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S0
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
FIELDNAME=DATE, ALIAS=, FORMAT=I6YMD,$
```

Issue a USE command for the temporary Master:

```
USE
ORIGDATE AS TMPDATE
END
```

5. Write a MODIFY to perform the conversion:

```
MODIFY FILE TMPDATE
COMPUTE DATEYMD/YMD=; DATEINT/I8=;
NEXT DATE
ON NEXT COMPUTE DATEYMD = D.DATE;
ON NEXT COMPUTE DATEINT=DATEYMD;
ON NEXT COMPUTE DATE =DATEINT;
ON NEXT UPDATE DATE
ON NONEXT GOTO EXIT
DATA
END
```

6. Rebuild the database with REBUILD REORG.
7. Test the new database using the new Master.
8. Delete the original database.
9. Delete the temporary Master.
10. Rename the new database and Master.

A6YMD Non-Key Field Conversion Process

1. Copy the original master and save it to a new name.
2. Change the old date formats to Smart Date formats, adding filler fields following the date fields.

```
FILENAME=ORIGDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
FIELDNAME= DATE, ALIAS=, FORMAT=A6YMD,$FILENAME=NEWDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
FIELDNAME= DATE, ALIAS=, FORMAT=YMD,$
FIELDNAME= FILL, ALIAS=, FORMAT=A2,$
```

3. Copy the original database and save it to a new name.

FOCUS for IBM Mainframe

4. Copy the original Master to create a temporary Master. Identify the date field as a GROUP field and retain the original format. Add a temporary field with the Smart Date format as the first field, followed by a filler field with a format of A2 as the second field.

```
FILENAME=TMPDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
GROUPNAME=DATEG, ALIAS=, FORMAT=A6YMD,$
FIELDNAME=DATE, ALIAS=, FORMAT=YMD,$
FIELDNAME=FILL, ALIAS=, FORMAT=A2, $
```

Issue a USE command for the temporary Master:

```
USE
ORIGDATE AS TMPDATE
END
```

5. Write a MODIFY to perform the conversion. Blank out the filler field:

```
MODIFY FILE TMPDATE
NEXT NUM
ON NEXT COMPUTE DATE=D.DATEG;
ON NEXT COMPUTE FILL= ` `;
ON NEXT UPDATE DATE FILL
ON NONEXT GOTO EXIT
DATA END
```

6. Rebuild the index if required.
7. Test the new database using the new Master.
8. Delete the original database.
9. Delete the temporary Master.
10. Rename the new database and Master.

A6YMD Key Field Conversion Process

1. Copy the original Master and save it to a new name.
2. Change the old date formats to Smart Date format, adding filler fields following the date field:

```
FILENAME=ORIGDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=A6YMD,$
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
FILENAME=NEWDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=YMD,$
FIELDNAME=FILL, ALIAS=, FORMAT=A2, $
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
```

3. Copy the original database and save it to a new name.

4. Copy the original Master to create a temporary Master. Identify the date field as a GROUP field, retaining the original format. Change the SEGTYPE=S1 to S0. Add a temporary field with the Smart Date format as the first field, followed by a filler field with a format of A2 as the second field:

```
FILENAME=TMPDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S0
GROUPNAME=DATEG, ALIAS=, FORMAT=A6YMD,$
    FIELDNAME=DATE, ALIAS=, FORMAT=YMD,$
    FIELDNAME=FILL, ALIAS=, FORMAT=A2,$
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
```

Issue a USE command for the temporary Master:

```
USE
ORIGDATE AS TMPDATE
END
```

5. Write a MODIFY to perform the conversion. Blank out the filler field:

```
MODIFY FILE TMPDATE
NEXT DATEG

ON NEXT COMPUTE DATE = D.DATEG;
ON NEXT COMPUTE FILL= ' ';
ON NEXT UPDATE DATE FILL
ON NONEXT GOTO EXIT
DATA END
```

6. The file must be rebuilt (REBUILD REORG). Sample REBUILD Procedure:

```
REBUILD
    REORG
    DUMP
    ORIGDATE
    NO
REBUILD
    REORG
    LOAD
    NEWDATE
```

7. Test the new database using the new Master.
8. Delete the original database.
9. Delete the temporary Master.
10. Rename the new database and Master.

A6MDY AS A KEY FIELD

1. Copy the original Master and save it to a new name.
2. Change the old date formats to Smart Date format, adding filler fields following the date field:

```
FILENAME=ORIGDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=A6MDY,$
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
FILENAME=NEWDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S1
FIELDNAME= DATE, ALIAS=, FORMAT=MDY,$
FIELDNAME= FILL, ALIAS=, FORMAT=A2,$
FIELDNAME= NUM, ALIAS=, FORMAT=I5,$
```

3. Copy the original database and save it to a new name.
4. Copy the original Master to create a temporary Master. Identify the date field as a GROUP field, retaining the original format. Change the SEGTYPE=S1 to S0. Add a temporary field with the Smart Date format as the first field, followed by a filler field with a format of A2 as the second field:

```
FILENAME=TMPDATE, SUFFIX=FOC
SEGNAME=ONE, SEGTYPE=S0
GROUPNAME=DATEG, ALIAS=,FORMAT=A6MDY,$
FIELDNAME=DATE, ALIAS=, FORMAT=MDY,$
FIELDNAME=FILL, ALIAS=, FORMAT=A2,$
FIELDNAME=NUM, ALIAS=, FORMAT=I5,$
```

Issue a USE command for the temporary Master:

```
USE
ORIGDATE AS TMPDATE
END
```

5. Write Modify to perform the conversion. Blank out the filler field:

```
MODIFY FILE TMPDATE
NEXT DATEG
ON NEXT COMPUTE DATE = D.DATEG;
ON NEXT COMPUTE FILL= ` `;
ON NEXT UPDATE DATE FILL
ON NONEXT GOTO EXIT
DATA VIA FIDEL
END
```

6. The file must be rebuilt (REBUILD REORG). Sample REBUILD Procedure:

```
REBUILD
  REORG
  DUMP
  ORIGDATE
  NO
REBUILD
  REORG
  LOAD
  NEWDATE
```

7. TEST the new database using the new Master.
8. Delete the original database.
9. Delete the temporary Master.
10. Rename the new database and Master.

Robert Freeman also covers date conversion in his article, "Converting Old Database Date Formats to Smart Dates," which appears in the July 1991 issue of *Information Builders Systems Journal*.

Changing the Century That Was Implicitly Stored With Smart Dates

Even with the availability of Release 7.0.8R, there is still the problem of Smart Dates that stored an implied century in FOCUS files. Suppose you have a FOCUS file that includes projected graduation dates. The field, GRAD_DATE, has a format of MDY in the Master File description. Before we discovered that there was a problem, many people were added with graduation dates in the early 1900's that should have been in the early 2000s. While adding DEFCENT and YRTHRESH to the code will create new additions to the file, it will not remediate existing data. A REBUILD will not do anything, as the data was loaded into a Smart Date, and FOCUS will assume it was correct. So, what is the solution? You can manually change those dates via UPDATE. For example:

```
TABLEF FILE STUDENTS
PRINT GRAD_DATE BY STUDENT_ID
WHERE GRAD_DATE LE 'JAN 01 1950'
ON TABLE HOLD FORMAT ALPHA
END
```

Note: This format must be ALPHA so that the century portion does not get saved into the HOLD file. Make sure DEFCENT and YRTHRESH are set for the GRAD_DATE field when reloading the data into the database.

```
MODIFY FILE STUDENTS
FIXFORM FROM HOLD
MATCH STUDENT_ID
ON MATCH UPDATE GRAD_DATE
ON NOMATCH REJECT
DATA ON HOLD
END
```

FOCUS for IBM Mainframe

This brings up another issue. In this STUDENT file, there is a field for the date of birth that also has a format of MDY. Why didn't we do the same for that field? Here, a value of 1900 as a date of birth might be reasonable. We may not want to take the time to change the data in this case to store the year 2000. Currently we have no students enrolled that are born in the year 2000. So, changing the DEFCENT and YRTHRESH without changing the data here would be acceptable. Even with DEFCENT 19, YRTHRESH 00, we will not encounter problems until someone born in 2000 is 16 years old.

Application Changes in FOCUS for IBM Mainframe

One of the advantages of converting the application instead of the data is that all of the application does not have to be changed at the same time. Changes can be made on an application by application basis. Some programs will require no conversion or just minor adjustments. Finally, converting the application does not preclude converting the data at some future time.

Some of the disadvantages of converting the application are that some programs will require extensive code changes. Programs that handle dates in a range of more than 100 years will have to have eight-digit dates incorporated into their internal design. Determination of the correct value for the century will vary by context; this means each field will need to be individually evaluated for the 100-year span. Selection criteria in ad hoc reports may fail. Extract files may lose century information they need for subsequent reporting.

Converting From Traditional Dates to Smart Dates

This section will cover the basics of converting an application from traditional dates to Smart Dates. The first thing to keep in mind is that the Smart Dates must be in the same general format as the old dates. For example, you cannot go from I6YMD to MDY format. This conversion can be done using a COMPUTE or DEFINE. Let's look at some examples.

In the MFD, consider the following fields:

```
FIELD = DATE_I4YM, , I4YM,$
FIELD = DATE_I6YMD, , I6YMD,$
FIELD = DATE_A6YMD, , A6YMD,$
FIELD = DATE_I6, , I6,$
FIELD = DATE_A8, , A8,$ (i.e. 99/12/31)
FIELD = DATE_A10, , A10,$
```

What Is Remediation?

The chart shows what format you are converting from and to, and how to code it in a COMPUTE or DEFINE.

Convert From:	Convert To:	Code:
I4YM	YM	NEW_I4YM/YM=DATE_I4YM;
I6YMD	YMD	NEW_I6YMD/YMD=DATE_I6YMD;
A6YMD	YMD	NEW_A6YMD/YMD=DATE_A6YMD;
YMD	MDY	NEWDATE/MDY = NEW_A6YMD;
I6 or A6	MDY	OLD_DT/I6MDY = I6DT; SMRT_DT/MDY = OLD_DT;
A8	YMD	OLD_YMD/A6YMD = EDIT(A8DT, '99\$99\$99'); SMRT_YMD/YMD = OLD_YMD;
A10	QY	OLD_A8/A8YYMD=EDIT(A10DT, '9999\$99\$99'); SMRT_YYMD/YYMD = OLD_A8; SMRT_QY/QY=SMRT_YYMD;

When converting alphanumeric format dates into Smart Dates, FOCUS cannot convert a value of blank. This will cause a FOC200 to occur. SET ALLOWCVTERR=ON will not work for COMPUTED fields since an ACTUAL to USAGE conversion does not occur in this situation. You will need to use an IF condition that inserts a zero into the new date when the old date is blank. For example:

```
COMPUTE NEWDATE/YMD = IF OLDDATE EQ ' ' THEN 0 ELSE OLDDATE;
```

or

```
COMPUTE TEMPDATE/I6YMD = EDIT(OLDDATE);  
NEWDATE/YMD = TEMPDATE;
```

What if you are not using Smart Dates?

As you know by now, Smart Dates are both convenient and useful. However, for various reasons, there are many applications that are still using conventional/legacy date formats such as A6YMD, I6MDY and so on. Keep in mind that if you convert conventional dates to Smart Dates, FOCUS enforces automatic validation. FOCUS will not accept dates with values such as 99/99/99 or 00/00/00, as they do not contain valid months or days. Note that there are applications that use a 13-month calendar, or a constant 30-day month. Dates generated by these applications would not be good candidates for Smart Dates, as they would be rejected on input. In Noreen Redden's article from the special Year 2000 issue of *Information Builders Systems Journal*, September/October 1998, entitled "I Never Promised You a Rose Garden (or When Smart Dates May Not Work)," describes some situations where converting to Smart Dates may not be the best option.

Release 7.0.8R, features a SET parameter, SET ALLOWCVTERR = ON/OFF.

- If ALLOWCVTERR=OFF, invalid dates generate a format conversion error.

FOCUS for IBM Mainframe

- If ALLOWCVTERR=ON with MISSING=ON, invalid dates are considered MISSING.
- If ALLOWCVTERR=ON with MISSING=OFF, invalid dates are converted to the base date.

Dialogue Manager is another reason to continue to use old date formats. In Dialogue Manager routines, you cannot specify formats with &variables. Thus the need for user-written subroutines (AYMD and the like) and functions (YMD, DMY, and MDY). Let's look at user-written subroutines first, and then at the functions.

User-Written Subroutines

As of Release 7.0.6 of FOCUS for IBM Mainframe, all user-written subroutines (with the exception of DTxxx and DAXxx) were changed to accept and deal correctly with eight-digit dates, or the equivalent. (Users of Release 7.0.8 and above, see the next paragraph.) AYM, which allows us to add a single month, accepts a six-digit input (CCYYMM), and creates a six-digit output. DAXxx subroutines were intended for, and are documented as, representing the number of days elapsed since 12/31/1899. The DTxxx subroutines take the number of days elapsed since 12/31/1899 and create a Gregorian date. These were not changed to accept eight-digit dates, and users who want full eight-digit date functionality are expected to convert to Smart Dates. Please note, however, that users will have to account for the fact that DAXxx and DTxxx have a different base date (12/31/1899) than the Smart Dates, which have a base date of 12/31/1900.

However, since there was a need for default century and year threshold awareness even for traditional dates utilizing the subroutines, Information Builders was asked to extend the "default century" rules to traditional dates even if only six digits were supplied and stored. This functionality of using DEFCEM and YRTHRESH with subroutines and functions was made available in Release 7.0.8. In 7.0.8R, the century will be included even if it was not provided when the date was input and if the output is large enough to display it. Please note that with the exception of CHGDAT, a subroutine cannot create more digits than it is supplied, but it is century aware. For instance, consider the following:

```
SET DEFCEM = 19, YRTHRESH=50
DEFINE FILE CAR
  OUTPT1/I6YMD WITH COUNTRY = AYMD(000228,1,'I6YMD');
  OUTPT2/I8YMD WITH COUNTRY = AYMD(000228,1,'I8YMD');
  OUTPT3/YMD WITH COUNTRY = OUTPT1;
END
```

In releases prior to 7.0.8R, this code will give the correct result for OUTPT1: 00/02/29 (2000 is a leap year), but note the results for OUTPT2 and OUTPT3:

```
OUTPT2 : 0000/02/29 -- century digits are not supplied
OUTPT3 : 2000/02/29 -- century digits are supplied by DEFCEM,YRTHRESH
```

In release 7.0.8R, this code will give the correct result for OUTPT2: 2000/02/29.

Note: The FOCUS user-written subroutines are documented in the *FOCUS for IBM Mainframe User-Written Subroutines Library* (DN1000026.0398). Please be sure that you have the latest version of this manual.

In releases prior to 7.0.8R, the TODAY subroutine only supported an output of 'A8'. In Release 7.0.8R, the subroutine TODAY was changed to support an output of either 'A8' (mm/dd/yy), or 'A10' (mm/dd/yyyy).

Functions

Currently, we have three functions that subtract one traditional date from another. They are YMD, MDY and DMY. If the fields were Smart Date formats, subtraction would be simple. However, there are applications that contain legacy dates or employ the use of Dialogue Manager. In Release 7.0.6, only the YMD subroutine works for eight-digit numeric dates, which have a format of I8YYMD. The MDY and DMY functions do not work with eight-digit numeric and alpha dates until Release 7.0.8. In Release 7.0.8, these functions are DEFCENT and YRTHRESH aware. So, with DEFCENT=19, YRTHRESH=50, you can subtract 12/31/99 from 00/01/01 and get the correct answer.

In Release 7.0.8R, there are four new date functions to be used in conjunction with two new calendar-setting features: SET BUSDAYS and SET HDAYS.

SET BUSDAYS =

Defines the days of the week to be considered business days.

SET HDAYS =

Allows you to declare a list of holiday dates for current, future and past years.

DATEADD

Adds months, days, years, business days, and weekdays to a Smart Date.

DATEMOV

Snaps a Smart Date to the beginning or end of a period, such as the start or end of the month, and many other useful points in time.

DATEDIF

Takes the difference between two Smart Dates.

DATECVT

Converts any date format to any other format.

DATEADD, DATEMOV, and DATEDIF can only be used with Smart Dates. DATECVT can be used with any date format.

Documentation on these new date functions can be found on the FOCUS 7.0.8R Documentation CD, DN100100.0698. For greater detail on how to use these new features, read “New Functions for Smart Dates,” by Joel Starkman, in the special Year 2000 issue of *Information Builders Systems Journal*, September/October 1998, Volume 11, Numbers 4 and 5, page 96. Also in the same issue, on page 149, is an article entitled “Testing Date Subroutines,” by Gregory Kiefer, which provides sample code for a utility that tests date subroutines.

Displaying the Date

Consider the way in which the data is being displayed on the report. Is it ambiguous? Suppose the report displays a field value of 01/31/00. Is the full date January 31, 1900, or is it January 31, 2000? Sometimes, the context of the field enables a user to determine which is the correct date. For instance, if the value is a projected graduation date, then we can assume that the century is 2000, since we may not care who graduated on 1/31/1900. However, what if this is a date of birth field? In a few years, it will be quite possible that a database could be keeping track of people born in the year 1900 and year 2000, so you will need to be able to make that distinction. In a case like this, a 100-year span is not a wide enough range for that date field, and it is not a good candidate for DEFCENT and YRTHRESH. In this case, you might consider expanding the field to the full eight-digits.

Sorting

Sorting is a problem when the span of two-digit years in a database crosses the century boundary. The sort order is reversed, and 00 which represents the Year 2000 will be sorted prior to 99, which represents the Year 1999. The same is true for traditional dates with two-digit years. To a sort function, traditional dates are only digits. Suppose you have a student file that stores expected graduation dates in I6YMD format. By redefining the field to an 'I9' format, we can see the actual value that is stored in the database. By printing this value, it shows us why the report is sorting the way it does. The second column, GRAD_DATE, has as a value of 602, which appears first in the sort because it sequentially comes before the value of 990603.

```
TABLE FILE STUDENTS
PRINT GRAD_DATE/I9 STUDENT_NAME BY GRAD_DATE
END
```

GRAD_DATE	GRAD_DATE	STUDENT_NAME
00/06/02	602	ADAMS, ABAGAIL
01/06/01	10601	BENJAMIN, BENJAMIN
99/06/03	990603	CARSTAIRS, CHARLES

These are not the results we were expecting. In order to get the expected output we can redefine the field, GRAD_DATE, to a YMD format, supplying DEFCENT and YRTHRESH.

```
DEFINE FILE STUDENTS
GRAD_DATE_2/YMD DFC 19 YRTHRESH 50 = GRAD_DATE;
END
```

```
TABLE FILE STUDENTS
PRINT GRAD_DATE_2/I9 STUDENT_NAME BY GRAD_DATE_2
END

GRAD_DATE_2          GRAD_DATE_2*          STUDENT_NAME
99/06/03             35948             CARSTAIRS, CHARLES
00/06/02             36313             ADAMS, ABIGAIL
01/06/01             36677             BENJAMIN, BENJAMIN
```

*This shows the offset from the base date.

External Sorts

If the data is being passed to an external sort utility, that sort will not interpret FOCUS's sliding window commands. In either situation, you must redefine the field to be a Smart Date and do the sorting based on the Smart Date field. Smart Date fields will be sorted properly with FOCUS's internal sort utility and by the external sort utility.

Selection Tests (on date fields)

When you compare two date fields, make sure that you are not mixing formats. For example, you can not compare an alphanumeric field to a numeric field, a two-digit year to a four-digit year, a YM field to a Y field, or a YMD field to a YM field.

Suppose you have a date field in the format of MDY and you want to find out if this is a payday (the 1st or the 15th of the month). You cannot just compare the value in the MDY field to '1' and '15'. You will never get a match. First, you would have to move the MDY field into a D field. Then you can then compare the value in the D field to the value of '1' or '15'.

The level of detail is not available with every format. Lower detail levels are lost going to a higher level. For example, if you have a MDYY field with the value of '12/31/1999' and it is converted to a MYY field, the value becomes '12/1999'. The day portion is lost and is not retained. If you go from a higher to lower level, it converts from a starting point. For example, when an MY field, with a value of '12/99' is converted to a MDY field, the value becomes '12/01/99'. This point is even more noticeable when converting from a QY field to an MDY field. For the QY field, you have the value of 'Q4 99'. This converts to a MDY field with a value of '10/01/99', which is the first month of the fourth quarter.

Smart Dates

When screening dates that use a date literal, the order of the components (month, day, and year) must match the format of the field. For example, HIRE_DATE is a field with the format of YMD. A screening on this field could look something like this:

```
WHERE HIREDATE EQ '19901215'
```

Note that the date literal is in quotes. If no quotes are supplied, FOCUS will interpret it as the internal storage value of the offset from the base date:.

FOCUS for IBM Mainframe

```
WHERE HIREDATE EQ 32856
```

Another option is to use natural date literals. This means that the month is in letters and the day and full year can be in any order.

```
WHERE HIREDATE EQ 'DEC 15, 1990'
```

All three of these WHERE tests will return the same results.

Traditional Dates

Consider the output from the following request:

```
TABLE FILE STUDENTS
PRINT STUDENT_NAME BY GRAD_DATE
END
```

Output:

GRAD_DATE	STUDENT_NAME
00/06/02	ADAMS, ABIGAIL
01/06/01	BENJAMIN, BENJAMIN
99/06/03	CARSTAIRS, CHARLES

If you are testing the following code on an A6YMD or I6YMD field you get 0 records returned:

```
TABLE FILE STUDENTS
PRINT STUDENT_NAME BY GRAD_DATE
WHERE GRAD_DATE FROM '990101' TO '020101'
END
NUMBER OF RECORDS IN TABLE =          0          LINES=          0
```

Nothing meets this selection test. You have to redefine the field, GRAD_DATE, to a Smart Date using DEFCENT and YRTHRESH.

```
DEFINE FILE STUDENTS
GRAD_DATE_2/YMD DFC 19 YRTHRESH 50 = GRAD_DATE;
END
TABLE FILE STUDENTS
PRINT STUDENT_NAME BY GRAD_DATE2
WHERE GRAD_DATE2 FROM '990101' TO '020101'
END
NUMBER OF RECORDS IN TABLE=          3  LINES=          3
```

GRAD_DATE_2	STUDENT_NAME
99/06/03	CARSTAIRS, CHARLES
00/06/02	ADAMS, ABIGAIL

Relational (SQL) Files

At first, it might seem that SQL files are not vulnerable to the Year 2000 problem because relational databases support elapsed date formats. However, because FOCUS allows write access to relational files, it is worth closer look. If the Master File description contains the settings ACTUAL=DATE, and USAGE=YYMD, MDYY or DMY for a date datatype column in a relational table, the only consideration would be to make sure that a four-digit year (not a two-digit year) is being used during data processing and data entry. Please refer to the sections on Data Processing and Data Entry for more details. If you have write access to the SQL file, and the Master File description contains ACTUAL=DATE, and USAGE=YMD, MDY, or DMY, you might want to add DEFCENT and YRTHRESH to correctly store the century portion. However, if FOCUS only has READ access to the SQL file, you will want to consider how these date fields are displayed. This is discussed in the Data Display section.

Non-Relational (Non-SQL) Files

In situations where a date value in the non-relational database does not include the century, that is, the ACTUAL is a six-digit integer or alphanumeric date value and the USAGE format is a Smart Date, FOCUS uses 1900 as the default century.

FOCUS cannot change the actual storage of date fields in non-relational files. The date formats and storage are predetermined by the programming language and database used to create the file. The date fields can contain a full four-digit year or a two-digit year. However, since the Master File description is used to describe the data to FOCUS, we can control how we want values to display in FOCUS by assigning the desired format in the USAGE parameter in the Master File. The Master File description uses two data formats, ACTUAL and USAGE, to describe the data in external (non-FOCUS) files. The ACTUAL specifies how the data is formatted/stored in the external file, and the USAGE specifies how it will be displayed by FOCUS. When data is read, it is converted from its external format (the ACTUAL format) to a FOCUS format (the USAGE or FORMAT=format). In this case, DEFCENT and YRTHRESH can be applied when the date value contains only a two-digit year.

Dialogue Manager and &Variables

In the absence of a specified format, Dialogue Manager mandates the use of subroutines and functions for conventional dates. As of Release 7.0.6, there is a new set of variables representing today's date in the next century. These new amper variables are required for working with Dialogue Manager, since DEFCENT and YRTHRESH have no impact on &DATE variables.

- &DATEYYMD is today's date shown as yyyy/mm/dd
- &DATEMDYY is today's date shown as mm/dd/yyyy
- &DATEMTrDYY is today's date shown as Month dd yyyy

FOCUS for IBM Mainframe

In Release 7.0.8R, &YYMD, &MDYY and &DMYY were added.

Note: When you are using date &variables in a DEFINE, you need to put single quotes around them or else you will get a “0” or blanks as the output. In addition, you can do “Smart Date” arithmetic in Dialogue Manager, as long as you present Dialogue Manager with the variables in “natural form,” rather than digits.

```
-SET &STRT= EDIT(&MDY,'9999' | 19 | EDIT(&MDY,'$$$$99');  
-SET &FROM = 01011997;  
-* when dealing with numbers in dialogue manager,  
-* remember to concatenate a non-numeric  
-SET &STRTR = CHGDAT('MDYY ', 'MDYYX ', (&STRT | 'X'), 'A17');  
-SET &FRTR = CHGDAT('MDYY ', 'MDYYX ', (&FROM | 'X'), 'A17');  
-SET &DIFF = &STRTR - &FRTR;  
-TYPE &|STRT IS &STRT: &STRTR  
-TYPE &|FROM IS &FROMDT: &FRTR  
-TYPE &|DIFF IS &DIFF:
```

Output:

```
&STRT IS 11301998: NOVEMBER 30 1998  
&FROM IS 01011997: JANUARY 01 1997  
&DIFF IS 698
```

Working With Non-Year 2000 Capable Releases

What if DEFCENT and YRTHRESH are not available?

Upgrade! Information Builders urges users operating in releases of FOCUS for IBM Mainframe that do not feature the DEFCENT and YRTHRESH commands to upgrade to Release 7.0.8R or above.

But if it is not possible to upgrade to the latest release, the first step is to determine the formats of all date fields. Are they currently storing the century? If the date fields store four-digit years and no date subroutines or functions are being used, and no calculations are being done, your application may already be Year 2000 compliant.

However, if your program doesn't meet these criteria, an alternative may be to code your own sliding window. Here's an example that converts data from an A6MDY format to A8MDYY and I8MDYY formats:

```

-SET &&YRTHRES = 'nn';
-SET &&DEFCENT1 = 19;
-SET &&DEFCENT2 = 20;
NEW_DATE_1/A8MDYY = IF OLD_DATE EQ ' ' THEN ' ' ELSE
                    IF OLD_DATE EQ '999999' THEN '99999999' ELSE
                    IF EDIT(OLD_DATE, '$$$$99' GE '&&YRTHRES' THEN
                    EDIT(OLD_DATE, '9999$$') | &&DEFCENT1 | EDIT(OLD_DATE,
                    '$$$$99') ELSE
                    EDIT(OLD_DATE, '9999$$') | &&DEFCENT2 | EDIT(OLD_DATE,
                    '$$$$99');
NEW_DATE_2/I8MDYY = EDIT(NEW_DATE_1);

```

Another option is to convert any old dates to Smart Dates. All the FOCUS User Written Date Subroutines were written prior to the existence of FOCUS Smart Dates in order to manipulate old dates. Smart Dates do every type of conversion that the subroutines can do.

The following examples show how you can get the same results, using different methods.

Using the DAxxx (Date) Subroutines:

The example below uses the DAxxx subroutine, which converts each date to the corresponding number of days elapsed since December 31, 1899. In FOCUS for IBM Mainframe, December 31, 1889 is Day 0. By converting dates into the numbers of days, you may perform arithmetic operations, such as adding and subtracting dates, and calculating the intervals between them. Then you may reconvert the results to a date format with the DTxxx subroutine. The DTxxx subroutine converts the number of days representing the date since December 31, 1899 to corresponding dates.

In this example, the DAYMD uses the old date format field, PAY_DATE with a format of I6YMD, as the input field and converts it into the number of days since Day 0. It adds seven to that integer to advance the date by one week, and then the DTYMD subroutine is used to convert that integer back to an old date format.

```

DEFINE FILE filename
IDATE/I5 = DAYMD(PAY_DATE, IDATE);
IDATE/I5 = IDATE +7;
NEXTWEEK/I6YMD=DTYMD( IDATE, NEXTWEEK );
END

```

This method crosses months and years, and addresses leap years until you cross the century boundary. These routines stop working when any integer value exceeds 36,524, which is the number of days in the 20th century.

Using Smart Dates

This method accomplishes the same task as the DAYMD subroutine but instead uses Smart Dates. Here, the current century for the old date format field, PAY_DATE, is first calculated using the sliding window technique. Then you can convert the old date field to a Smart Date field and add seven.

FOCUS for IBM Mainframe

```
DEFINE FILE filename
CC/A2 = IF EDIT(PAY_DATE, '99') GE '10' THEN '19' ELSE '20';
TDATE/A8YYMD = CC || EDIT(PAY_DATE);
SDATE/YYMD = TDATE;
NEXTWEEK/YMD=SDATE+7;
END
```

This method is valid for crossing the century boundary, as well as addressing leap years, months and so on. This is valid until 12/31/9999.

See the section, (REVIEWER: Which chapter contains this section?) *Smart Dates in FOCUS Files* to read about the many benefits of using Smart Dates. Smart Dates can be used in arithmetic-type operations, keeping track of dates, sorting data properly, performing automatic validation, and displaying data in a variety of formats.

Limitations of FOCUS Date Handling

Although Information Builders' sliding window technique, using DEFCENT and YRTHRESH, was designed to handle dates in the Year 2000 and beyond, date problems can still arise.

- If you have dates stored in an old date format, they are not validated unless the application specifically invokes validation. In other words, no testing is done to determine if the data represents actual dates. DEFCENT and YRTHRESH will not catch these errors. However, FOCUS Smart Dates will automatically validate the data, and the data can then be corrected and changed.
- DEFCENT and YRTHRESH will not be in effect for sorting. Two-digit year fields must be redefined to contain the century portion for correctly ordered sorting to occur.
- DEFCENT and YRTHRESH only handle a 100-year span. If the field requires a greater span, then you should consider expanding the field to a full eight-digit date. In Release 7.0.8R, there is an enhancement made to YRTHRESH so that the sliding window is a moving 100-year span. In addition to specifying a year, you can now set YRTHRESH as an offset from the current year. This technique creates a moving century window that increments itself each year without modifying your application. You decide the number of years to offset in YRTHRESH.
- DEFCENT and YRTHRESH will not change data that is already stored in the file.
- Data passed to an external sort will not interpret 'sliding window' commands.

Testing on FOCUS for IBM Mainframe

Many people underestimate the importance of testing. Make testing a priority, and coordinate testing with the conversion schedule. Before testing can even begin, you will need the following:

- Adequate time
- Adequate staffing
- A set of software testing tools
- A hardware and software test environment

Some of these elements will be discussed below. With adequate planning, the conversion process should go smoothly, but be sure to include a contingency plan and enough time to implement changes if testing shows that the remediation is not successful.

Staffing

Software testing requires a skilled and experienced staff. If your organization does not have a formal QA group, testing roles and responsibilities should be formalized, and a Project Leader assigned to the task. Proper tools should be acquired and staff training should be completed before the actual testing begins. The application programmers who made the changes should test the modified applications. The end users should also be made an integral part of the testing, as they can provide real scenarios on how the application is actually being used.

Environment

Release 7.0.8R, offers a new feature, SET TESTDATE, which is available specifically for testing future and past dates. For example, this can be set in a global profile to make it appear that the actual system date is a date in Year 2000. Ideally, your testing group will have an environment where an altered program can be tested without affecting the current production system. With MVS, a separate LPAR can be used or a dedicated machine can be IPL'd (initial program loaded) with the desired date. MVS users can also use an SVC11 screener product such as HourGlass 2000 or TICTOC.

FOCUS for IBM Mainframe

This test environment should contain copies of all the applications and data to be changed, as well as those that may be affected by the change. Compare the original program's output to the remediated program's output. The two should be identical. Then age the data by adding a set number of years to all the date fields. A utility can be used to perform the aging. Data aging is discussed in the article, "Testing Year 2000 Conversion," by John Patrick, in the September/October 1998 issue of *Information Builders Systems Journal*, Volume 11, Numbers 4 and 5, on page 71. Various system dates, such as those listed below, should be tested with the aged data. The testing environment should include batch jobs as well as online testing. Test your most important applications first, then your secondary ones. Be sure to remember month-end, end-of-quarter, and year-end jobs, as well as that one report that runs only when the CEO requests it.

Data

Contact each application's primary users to determine the best sampling of data to test on the revised application. Test a run against a copy of the production datasets, along with an aged copy of the data, to ensure that the modified code works correctly.

When testing with different centuries and calculations, keep in mind that the year 1900 was not a leap year, but 2000 is. Only centuries evenly divisible by 400, such as 1600 and 2000, are leap years. Many programs were designed with algorithms and will not recognize 2000 as a leap year. You will need to manually verify the results.

There are no industry standards for required test dates. Optimally, you would test every single day for each year prior to 2000, the year 2000, and after 2000. However, time and resources are limited. There are different dates to take into consideration for applications, hardware and systems software. The following section lists specific dates that should not be excluded.

Test Dates

Test Dates for Applications

December 31, 1998, Thursday

January 1, 1999, Friday

January 4, 1999, Monday

Processing these dates will test whether or not the Year 2000 changes that took place in 1998 will still work in 1999. Monday, January 4 is the first business day in 1999, just as Friday, January 1 is a holiday.

September 9, 1999, Thursday

In many systems, 99/99/99 and 09/09/99 are default values for missing dates. Make sure that these values occur only where intended.

December 31, 1999, Friday

This is the last day of 1999.

January 1, 2000, Saturday

The first day of 2000 is not a business processing day for most applications, but the data produced or processed on this day may be used on Monday, January 3 the first business day of 2000.

January 2, 2000, Sunday

As with January 1, January 2 is not a business day, but computers will process this date to prepare for Monday, January 3, the first business day of 2000.

January 3, 2000, Monday

This is either the first business day of 2000 or a bank holiday, depending on the country.

January 4, 2000, Tuesday

In some countries, this will be the first business day of 2000. In others, it will be the first online day following the first batch run of 2000.

January 5, 2000, Wednesday

If January 4th was the first business day for any part of the application processing, this will be the first online day following the first batch run of 2000.

January 31, 2000, Monday

The first month's end in 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

A leap day. Many systems fail every leap year on leap day due to lack of testing on the leap year determination code. A year is a leap year if divisible by 4, unless it is a century year, in which case it must be divisible by 400. For example 1700, 1800, and 1900 are not leap years, but 2000 is.

March 31, 2000, Friday

The first end-of-quarter in 2000.

January 2, 2001, Tuesday

The first business day of 2001, to ensure that the first year-end of 2000 works.

Test Dates for Hardware

These dates were chosen to verify that hardware components will retain the correct time and date after crossing over the Year 2000 boundary. Hardware should maintain current crossovers, such as 1998 to 1999.

December 31, 1998, Thursday

January 1, 1999, Friday

Testing the date crossover from year-end 1998 to the start of 1999.

December 31, 1999, Friday

January 1, 2000, Saturday

This is the crossover from year-end 1999 to the first day of 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

This is the crossover for the leap day in 2000.

Test Dates for Systems Software

Systems software includes operating systems, database management systems and so on. These dates were chosen to verify that the system software retains the correct time and date after crossing over to the Year 2000. System software should also maintain current crossovers from one year to the next.

December 31, 1998, Thursday

January 1, 1999, Friday

Testing the date crossover from year-end 1998 to the start of 1999.

December 31, 1999, Friday

January 1, 2000, Saturday

This is the crossover from year-end 1999 to the first day of 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

This is the crossover for leap day in 2000.

Testing Checklist

- Identify each application's resources.
- Identify target files for update, browse, etc.

- Generate initial test data to reflect a subset of production environment.
- Establish documentation procedures.
- Allow time for review of each application's results .
- Repeat process and roll out.

Testing Issues

Source Control and Management

- Minimize or forestall non-Year 2000 changes to production code during the remediation process.
- Track code prior to conversion.
- Establish a baseline and audit trail procedure.
- Formalize an approval and movement process.
- Automate deployment.
- Build in a contingency plan.
- Satisfy internal auditing.
- Develop a production code change procedure while code is in remediation.

Unit Testing Tasks

- Program and transaction level verification.
- Identify transaction flows for online applications.
- Black box testing, also known as functional testing.
- Keep test files and database sizes manageable.
- Create valid, invalid and unexpected test cases.
- Define what the acceptable criteria is for testing.
- Use the criteria as a building block for expanded testing.

Business Cycle Testing

- Date simulation during real time updates.
- End of month, quarter and fiscal year cycle.

FOCUS for IBM Mainframe

- Create data to trap timelines and totals.
- Consider running the current and test systems in parallel for one cycle.
- The test data set used here may be larger than the data set in other test cases.
- Encourage business analyst participation.

Test Coverage

- Determine how much code has been “exercised” during a specific test.
- Track untested lines of code.
- Calculate the percentage of code that has been tested.
- Make sure code that has been changed is tested.

Platform Specific Considerations

- On MVS, the minimum release level that IBM will certify as Year 2000 compliant is OS/390 Version 1.
- On VM, the minimum release level that IBM will certify as Year 2000 compliant is VM/ESA Version 2.

Other Considerations

- Always remember to analyze any HLI programs for date fields, since HLI programs are used to update FOCUS files.
- Assess the way in which date data imported from external systems is being described to FOCUS in the Master File description. Is the data stored with four or two-digit years?
- Be aware of the Year 2000 remediation efforts of other groups within your organization. It is easy to make changes that conflict with such efforts rather than work with them.
- Is your data being exported to other systems? Do exported dates contain the right century? These questions may also apply to HOLD files. If you perform an ON TABLE HOLD FORMAT ALPHA, the century portion does not get saved when holding Smart Date field formats of MDY, DMY, and YMD.

Summary of New Date Features

This section describes all the new date features. For more details, see the FOCUS 7.0.8R Documentation CD (DN1001050.0698), or the technical support section of www.ibi.com.

Note: Release 7.0.7 is not listed because there were no new date features in that release.

Release 7.0.5

Project 2000 – Phase I

SET DEFCENT=19

SET YRTHRESH=00

This is available at the global level only in this release.

Dialogue Manager enhancements for &DATE variables:

The syntax of &DATE is:

```
Set &DATEfmt = ;
```

Note: Although it is possible to display the century portion using &DATEfmt, if the system date is something other than '19,' then '19' will still be displayed. That is if the system date is 2000/01/31, and you use &DATEYYMD, the output will be 1900/01/31. This is corrected in release 7.0.7M.

Release 7.0.6

Project 2000 – Phase II

SET DEFCENT and SET YRTHRESH available at the file and field level.

FDFCENT/FDFC and FYRTHRESH/FYRT (file level).

DEFCENT/DFC and YRTHRESH/YRT (field level).

Release 7.0.7M

Acceptance of four-digit years in date subroutines.

Acceptance of century portion of operating system date (e.g. in amper variables).

Release 7.0.8

Project 2000 – Phase III

Subroutines will use DEFCENT and YRTHRESH, and the last argument has been expanded.

Date Handling for the Year 2000 in FOCUS:

The default date format display in FOCUS is now MMDDCCYY. This applies to all areas within FOCUS that display a date:

- The FOCUS Banner
- The FOCUS Banner
- The FOCUS Banner
- ? FILE
- ? FDT
- MODIFY FILE FN
- CREATE FILE FN
- FSCAN FILE FN
- REBUILD TIMESTAMP
- SET DATEFNS=ON/OFF

Focus functions: YMD, DMY, and MDY are now DEFCENT/YRTHRESH aware.

Release 7.0.8R

REBUILD / DATE NEW

Converts legacy dates to Smart Dates in Focus databases

SET DATEDISPLAY = ON/OFF

Controls display of base date: 12/31/00

CHECK FILE fn HOLD ALL

Writes DFC/YRT to Hold Master

ON TABLE HOLD

Writes DFC/YRT to Hold Master

New Date Math Functions for Year 2000:

DATEADD (smartdate, 'unit', #units)

SET BUSDAYS = {day-list/_MTWTF_}

SET HDAY= {filename} HDAY will prefix the filename. For example, if the filename is STKM, FOCUS will look for a file named HDAYSTKM with a filetype of ERRORS in VM or in the ERRORS PDS on MVS.

DATEDIF(FR_smartdate,TO_smart,'unit');

DATEMOV(smartdate, 'move-point', outdt);

DATECVT(date, 'oldate', 'newformat')

Displaying Invalid Smart Dates in Reports:

SET ALLOWCVTERR={ON|OFF}

OFF: invalid date gives format conversion error

ON: MISSING=ON – invalid date is MISSING

ON: MISSING=OFF – invalid date is base date

YRTHRESH = -n

YRTHRESH can now be set as an offset from the current year as well as specifying a year.

TODAY Subroutine enhancement

-SET &HELLO=TODAY('A10');

Display &Date variable without separators

&YYMD

&MDYY

&DMYY

YYJUL – Julian Format

SET TESTDATE={yyyymmdd/TODAY}

This allows you to set a date within FOCUS for testing purposes only.

MSO log changes

MSOPRINT log now displays with four-digit year.

Frequently Asked Questions

1. What is the Mega PTF?

The Mega PTF is a consolidation of more than 20 date-related fixes. This is a required PTF for Release 7.0.8.

2. Is the Mega PTF (PTF 87857 for MVS and PTF88008 for VM) that is required for 7.0.8 included in 7.0.8R, or is it also required for 7.0.8R?

The Mega PTF is applicable to, and required for, Release 7.0.8 **only!** Release 7.0.8R incorporates the fixes that are in the Mega PTF.

3. Where can I find the different features and fixes that are in releases 7.0.8 and 7.0.8R, as well as other 7.0 releases?

Go to <http://techsupport.ibi.com/support/ibm/toc.asp>. There is an area set up for new features and problems that are corrected in each release of FOCUS. This information is also available on the FOCUS 7.0.8R Documentation CD, DN 100100.0698.

4. What are DEFCENT and YRTHRESH?

Information Builders' solution to the problem of attributing a century to a date with a two-digit year field is to use the settings DEFCENT and YRTHRESH. These two new set parameters are used to define century values for a 100-year span of two-digit year dates.

5. How does setting DEFCENT and YRTHRESH affect using COMBINEs in the application?

The order prescribed by the DEFCENT/YRTHRESH rule still applies. FOCUS evaluates DEFCENT and YRTHRESH at the field level first, the file level next, and finally, the global level. If this is a COMPILED MODIFY, you will need to set YRTHRESH and DEFCENT to the same settings used when the module was compiled, before you can run the FOCCOMP module.

6. Are DEFCENT and YRTHRESH used in MODIFY procedures?

Yes. MODIFY will use DEFCENT and YRTHRESH settings. It can be set in four places.

1. It can be specified on the left-hand side of the "=" in a COMPUTE statement at the field level.
2. It can be specified at the field level in the Master File description.
3. It can be specified at the file level in the Master File description.
4. It can be specified globally.

7. Does MAINTAIN support DEFCENT and YRTHRESH?

Yes. As of release 7.0.8R MAINTAIN supports DEFCENT and YRTHRESH at the global, file, and field levels.

8. What field formats can I apply DEFCENT and YRTHRESH to?

DEFCENT and YRTHRESH can be applied to date fields that do not provide the century portion and where date conversion takes place. Date conversion takes place for Smart Dates only and evaluated only when the century digits are not supplied and the user is:

1. Converting old dates to Smart Dates.

2. Converting from ACTUAL to USAGE that is a Smart Date format.
3. Performing data entry via FIXFORM or CRTFORM.

You can also apply DEFCENT and YRTHRESH to traditional dates that lack century values and are being used as input to subroutines.

9. What happens if you code DEFCENT and YRTHRESH, but the field already contains the century?

If the century is explicit when assigning the value and when the Smart Date is already stored, DEFCENT and YRTHRESH will not be applied..

10. What does the 'R' stand for in release 7.0.8R?

The 'R' stands for remediation. Release 7.0.8R is considered the remediation release due to the new features that were included to assist users in the Year 2000 remediation process.

11. Is the Cobol FD Translator (CTF) Year 2000 compliant?

Year 2000 compliance is not a relevant issue for CTF. CTF outputs a Master File description from a COBOL FD. The COBOL FD does not contain formats reserved specifically for dates, so CTF cannot determine if a field is a date or not. Of course, the user can edit the generated MFD to change the FOCUS formats to DATE formats and then include the threshold values. But again, there is no intelligence built into the COBOL FD that will allow CTF (or any other tool) to do date identification in any language.

12. Does the FOCUS Graph facility work with Year 2000 dates?

As of release 7.0.9, Smart Dates will work with Graph. However, Graph will not work with eight-digit legacy date formats.

13. Can my application be Year 2000 compliant if I'm in a release prior to 7.0.8 and if I do not use any of the Date Subroutines?

Releases of FOCUS for IBM Mainframe prior to Release 7.0.8 can be used to develop Year 2000 compliant applications, so long as those applications include four-digit years, use date math that is century-aware, and do not include FOCUS date subroutines or FOCUS date arithmetic.

14. Why do I get the wrong value returned when I DEFINE a field to the system date?

For example, today's date is January 29, 1999 and you have the following DEFINE statement:

```
TODAY/YYMD = &DATEYYMD;
```

FOCUS for IBM Mainframe

The output displays a value of '1901/03/09'.

The problem here is that FOCUS takes the value of today's date, which in this case is '1999/01/29', and is dividing the numbers into each other with the result of 68. This value becomes the stored value for this DEFINEd Smart Date field, and the value displayed, '1901/03/09', is 68 days from the base date of 12/31/1900.

In order to get the expected result, the DEFINE statement must have quotes around the value so that it will be evaluated as a date literal. For example:

```
TODAY/YYMD = '&DATEYYMD' ;
```

4 EDA

Remediation is the process of modifying an application to make it capable of handling date values beyond 12/31/99. In a fully remediated application, the data is changed to contain unambiguous date values, the code is changed to correctly display, sort, select and store unambiguous dates, and the EDA Server and the operating system it relies on are upgraded to Year 2000 capable release levels.

Achieving Year 2000 compliance in a client-server environment like EDA might seem overwhelming. In this environment, data and applications can reside on many different machines of various architectures, as well as exist in a myriad of database types, and applications can be written in a host of different languages,

However, EDA has features and capabilities that can help minimize the impact of the Year 2000 problem. For example, EDA has built-in functionality for SQL-based EDA applications which provides default century values for many partial date container data types. EDA also has features that help address Year 2000 issues with non-SQL-based EDA applications.

Determining If You Have a Year 2000 Problem

The first step in determining whether or not you have a Year 2000 problem is to identify where and how date fields are being used within your environment. In many environments, it will be necessary to conduct several searches for any potentially ambiguous dates that might need to be remediated, replaced or removed. This task will involve examining all metadata, programs, reports, input and output screens, databases and any associated hardware and software on a single machine. In an environment like EDA, where several different types of applications (which may or may not handle their own date processing) may reside on many different types of machines, and where metadata, programs and data might reside on many remote machines of different architectures, the task of examining the environment can expand exponentially.

Determining Where to Remediate

In a client-server environment, making the decision on where changes should be made (on the client or server machines or both) might be as important as determining what to change. Making the change in the right place can potentially save time, money and effort. Making changes to the application on the client machine might seem to be a good short-term solution if the personnel needed to make the changes on the server machine are not available. However, if the application has to be rolled-out to many users, or if there are many different types of client applications that need to be changed, then the time, effort and expense that might be involved may not be justifiable. Remediation at the server level, instead of client application level, is a better long-term solution for consistency and ease of maintenance. If you remediate at the server level, all applications accessing the central server can take advantage of your changes as soon as the environment is recycled and the EDA Server administrator can better coordinate the change effort. In addition, if the client application was written by a vendor or third-party developer, then remediation efforts at the client end might be out of your control. What follows is a discussion of some other items to consider when deciding whether to remediate on the client or server machine.

When might remediation on the Client be necessary?

It might be necessary to remediate on the client machine when:

- All data, procedures and metadata on the EDA Server machine are already Year 2000 compliant and only the client applications' screens, reports or processing of dates need to be remediated for Year 2000.
- The staff involved in remediation does not have access to the server machine, libraries or data files.
- The personnel or other resources needed to remediate on the server are unavailable.

When is it better to remediate on the EDA Server machine:

It is better to remediate on the EDA Server machine when:

- All screens, reports and processing done by the client applications are already Year 2000 compliant and only the data, metadata or procedures on the server machine need to be remediated.
- The data accessible to the EDA Server is not being used for display-only purposes, or storage of the century with date columns is required.
- There are many stand-alone applications, or different types of applications, on one or many client machines that would need to be remediated in order to achieve Year 2000 compliance, and making the changes on the server is more efficient.
- The client application was written by a third party or vendor and cannot be altered.

- Century remediation of dates at the client is not feasible. For instance, if an SQL-based EDA application is used, the EDA Server will automatically return most dates in YYYYMMDD format providing a default century value for dates which do not contain a century component. In this instance, if remediation of the default century value is required, it would need to be performed at the server level. See Level 1: Application Considerations for more details about this topic.

Determining What to Remediate

The task of determining what to remediate in a client-server environment like EDA can seem overwhelming. To help simplify this task, we will examine what areas in an EDA environment should be considered for Year 2000 remediation. By dividing the task into different levels, we can also provide recommendations on what personnel might be needed to make changes as they become necessary. The levels are defined as follows:

	Level	Recommended Personnel
Level 1:	Application Considerations	Application Developer
Level 2:	EDA Server Considerations	EDA Server Administrator Application Developer
Level 3:	Database Considerations	EDA Server Administrator Database Administrator Application Developer

Level 1: Application Considerations

The application that is the front-end to your EDA environment will primarily be involved in presenting the data returned from the EDA Server, but it might also provide some processing functions of its own. Examination at the application level for Year 2000 considerations can be divided into three categories: Data Presentation Considerations, Data Processing Considerations and Other Considerations. Questions or concerns that might arise when examining these areas are discussed below.

Data Presentation Considerations:

How does EDA return date columns with and without a century?

The answer depends on a number of factors. The primary factor is whether the request received by the EDA Server is an SQL request or a TABLE request. The EDA Server release level, and whether the date value stored in the database contains an explicit, implicit or no century value, are other important factors. The next few paragraphs describe how these and other factors can affect the date value returned by EDA.

SQL SELECT requests

For SQL SELECT requests generated against EDA Release 3.3.2 (MVS) and Release 4.x (for other platforms including MVS) servers, all date columns will be returned in YYYYMMDD format regardless of the original format or whether the date value includes an explicit century. For SQL SELECT requests generated against EDA Servers Release 3.3.1 and below, the display format of the date will depend on whether the date is a numeric or alphanumeric date data type. For numeric date data types, all columns will be returned in YYYYMMDD format regardless of the original format or whether the date value includes an explicit century. For alphanumeric date data types, all columns will be returned according to the format (USAGE or FORMAT attribute) specified in the EDA Master File description, and the century value will only be displayed if it is explicitly available in the database. The chart below illustrates how dates are formatted in EDA Release 3.3.1 and below, and EDA Release 3.3.2 and above:

Numeric Date Formats	Display Format in EDA 3.3.1 and Below	Display Format in EDA 3.3.2 and Above
DATE data type in relational table	YYYYMMDD	YYYYMMDD
YYMD	YYYYMMDD	YYYYMMDD
YMD	YYYYMMDD	YYYYMMDD
YM	YYYYMMDD	YYYYMMDD
I8YYMD	YYYYMMDD	YYYYMMDD
I6YYMD	YYYYMMDD	YYYYMMDD
I4YM	YYYYMMDD	YYYYMMDD
MDYY	YYYYMMDD	YYYYMMDD
MDY	YYYYMMDD	YYYYMMDD
I8MDYY	YYYYMMDD	YYYYMMDD
I6MDY	YYYYMMDD	YYYYMMDD

DMY	YYYYMMDD	YYYYMMDD
DMY	YYYYMMDD	YYYYMMDD
I8DMYY	YYYYMMDD	YYYYMMDD
I6DMY	YYYYMMDD	YYYYMMDD
A8YYMD	YYYYMMDD	YYYYMMDD
A6YMD	YYMMDD	YYYYMMDD
A4YM	YYMM	YYYYMMDD
A8MDYY	YYYYMMDD	YYYYMMDD
A6MDY	MMDDYY	YYYYMMDD
A8DMYY	DDMMYYYY	YYYYMMDD
A6DMY	DDMMYY	YYYYMMDD

The following SQL scenarios illustrate in more detail how the factors mentioned above will affect the date values returned by EDA for SQL requests against specific date data types. The Year 2000 considerations for these data types are also discussed.

SQL Scenario 1: DATE Datatype Columns in SQL Passthru Mode - All EDA Releases

If the date columns are DATE data type columns in a relational table, and they are being retrieved from an EDA relational gateway server or in SQL passthru mode, then these columns will be converted to YYYYMMDD format upon output, and the explicit century values contained in the database will be returned.

The chart below illustrates how these columns would be returned by EDA:

DATE data type column value as returned by native relational reporting tool	Date value returned from an EDA relational gateway or SQL passthru mode request
2000-12-31	20001231
1999-12-31	19991231

Note: The date values returned by EDA might display with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used. The native relational reporting tool used in the above example was DB2's SPUFI.

What are the Year 2000 considerations for these types of columns?

An explicit century value is already available within the database for these columns. Therefore, as long as the work areas in your application that receive these columns, or the output screens and reports that display them, can already accommodate dates which include the century, no adjustments would be necessary to your display environment to achieve Year 2000 compliance. If an inaccurate date value is returned from a request in passthru mode or through an EDA relational gateway, the same SQL request should be issued in the native reporting tool for the relational database being accessed since EDA retrieves the value from the relational database.

SQL Scenario 2: Numeric Full Date Container Columns - All EDA Releases

If the date columns are full date containers, have numeric date formats (these date columns might be fields described in an EDA Master File description as I8YYMD, I8MDYY, I8DMYY, or YYMD), and are being retrieved from an EDA Hub or Full Function server, then these date columns will be converted to YYYYMMDD format upon output, and the explicit century values contained in the database will be returned.

The chart below illustrates how 20th and 21st century date values would be returned by EDA for these columns:

Date value	EDA Master File Usage Format	Value Returned by EDA
20001231	YYMD	20001231
19001231	YYMD	19001231
20001231	I8YYMD	20001231
19001231	I8YYMD	19001231
12312000	I8MDYY	20001231
12311900	I8MDYY	19001231
31122000	I8DMYY	20001231
31121900	I8DMYY	19001231

Note: The date values returned by EDA might display with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

What are the Year 2000 considerations for these types of columns?

Explicit century values are already available within the database for these types of columns. Therefore, as long as the work areas in your application that receive these columns, and the output screens and reports that display them, can already accommodate dates that include the century, no adjustments would be necessary to your display environment to achieve Year 2000 compliance.

SQL Scenario 3: Alphanumeric Full Date Container Columns - EDA Release 3.3.2 (MVS) & 4.x (All Platforms)

If the date columns are full date containers with alphanumeric date formats (these date columns might be fields described in an EDA Master File description as A8YYMD, A8MDYY, or A8DMYY), and they are retrieved from an EDA Hub or Full Function Server, then these fields will be converted to YYYYMMDD format upon output, regardless of the original formats in the Master File description. The explicit century values contained in the database will be returned.

The chart below illustrates how 20th and 21st century date values would be returned by EDA for these columns:

Date Value	EDA Master File Usage Format	Value Returned by EDA
20001231	A8YYMD	20001231
19001231	A8YYMD	19001231
12312000	A8MDYY	20001231
12311900	A8MDYY	19001231
31122000	A8DMYY	20001231
31121900	A8DMYY	19001231

Note: The date values returned by EDA might be displayed with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

What are the Year 2000 considerations for these types of columns?

Explicit century values are already available within the database for these types of columns. Therefore, as long as the work areas in your application that receive these columns, and the output screens and reports that display them, can already accommodate dates that include the century, no adjustments would be necessary to your display environment to achieve Year 2000 compliance. Date screening can be performed on these columns in YYYYMMDD format even if the EDA Master File description uses a different date component order. In other words, the order of the date components in the screening condition does not have to match the date component order specified in the EDA Master File description. In the example below, the following SQL request would return the desired date values:

```
FILE=TESTDB, SUFFIX=FIX,
SEGNAME=ONE,
FIELDNAME=A8MDYYFLD, ALIAS=A8MDYYFLD, USAGE=A8MDYY, ACTUAL=A8,$
```

Testdb Data:

```
A8MDYYFLD
12311900
```

EDA

SQL request with screening in YYYY-MM-DD format:

```
SELECT A8MDYYFLD  
FROM TESTDB  
WHERE A8MDYYFLD = '1900-12-31';
```

Results:

```
A8MDYYFLD  
19001231
```

SQL Scenario 4: Alphanumeric Full Date Container Columns - EDA Release 3.3.1 & below

If the date columns are full date containers with alphanumeric date formats (these date columns might be fields described in an EDA Master File description as A8YYMD, A8MDYY, or A8DMYY), and they are retrieved from an EDA Hub or Full Function Server, then these columns will be returned in the date component order specified by the USAGE or FORMAT attributes in the EDA Master File description. In other words, they will not be converted to YYYYMMDD format upon output. The explicit century values contained in the database will be returned.

The chart below illustrates how 20th and 21st century date values would be returned by EDA for these columns:

Date Value	EDA Master File Usage Format	Value Returned by EDA
20001231	A8YYMD	20001231
19001231	A8YYMD	19001231
12312000	A8MDYY	12312000
12311900	A8MDYY	12311900
31122000	A8DMYY	31122000
31121900	A8DMYY	31121900

Note: The date values returned by EDA might be displayed with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

What are the Year 2000 considerations for these types of columns?

An explicit century value is already available within the database for these types of columns. Therefore, as long as the work areas in your application that receive these columns, and the output screens and reports that display them, can already accommodate dates which include the century, no adjustments are necessary to your display environment to achieve Year 2000 compliance. However, date screening must be performed in the date component order specified by the USAGE or FORMAT attribute in the EDA Master File description and cannot include dashes between the year, month and day components of the date. Therefore, the SQL screening examples used in SQL Scenario 3 above would not return the expected date values in this instance. The following SQL screening would, however, return the desired date results:

```
FILE=TESTDB, SUFFIX=FIX,
SEGNAME=ONE,
FIELDNAME=A8MDYYFLD, ALIAS= A8MDYYFLD, USAGE=A8MDYY, ACTUAL=A8,$
```

Testdb Data:

```
A8MDYYFLD
12311900
```

SQL request with screening in MM/DD/YYYY format:

```
SELECT A8MDYYFLD
FROM TESTDB
WHERE A8MDYYFLD = '12311900';
```

Results:

```
A8MDYYFLD
12311900
```

SQL Scenario 5: Implicit Century Numeric Partial Date Container Columns - All EDA Releases

If the date columns are partial date containers that are stored in the database with an implicit century (these date columns might be fields described in an EDA Master File description with Smart Date formats such as YMD, MDY, DMY, YM, etc.), and are retrieved from an EDA Hub or Full Function Server, then these fields will be converted to YYYYMMDD format upon output. In addition, the implicit century value that was stored when the file was loaded will be returned. FOCUS databases and files that can be described to EDA in the Master File description with ACTUAL=DATE and a Smart Date USAGE format (i.e. HOLD files created from FOCUS files containing Smart Date columns), have the ability to store an implicit century. A FOCUS database field described with a Smart Date format, such as YMD, MDY or DMY, will be loaded with a value containing an implicit century.

EDA

The advantage of using Smart Date formats in this instance is that a Smart Date will store the date value internally as an integer representing the number of days since the base date of December 31, 1900, and will store either an explicit century (for YYMD and MDYY formats) or implicit century (for YMD, MDY, etc. formats). Because of this, accurate date math can be performed on these columns across centuries, with leap years taken into account. Sorting and screening operations will also function properly on these columns. For date types such as YM, which do not include a day component, the date value will be returned in YYYYMMMD format using the implicit century, and a default value of the first day of the month will be assigned for the day component. The chart below illustrates this:

Date Value	EDA Master File Usage Format	Value Returned by EDA
20001231	YMD	20001231
19001231	YMD	19001231
12312000	MDY	20001231
12311900	MDY	19001231
31122000	DMY	20001231
31121900	DMY	19001231
200012	YM	20001201
190012	YM	19001201

Note: The date values returned by EDA might be displayed with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

What are the Year 2000 considerations for these types of columns?

Even though FOCUS databases have the ability to store an implicit century, consideration still has to be given to how the date values were entered into the database. If the data entry to these columns only permits a user to enter the last two-digits of the year, then a default century value of '19' is probably stored in the database. The implications of this situation, and remediation options, are explored in the Level 3 - Database Considerations section.

Screening on these date fields is also a consideration. The century must be included as part of the screening criteria when screening on these dates in an SQL SELECT statement. Otherwise, '0' records or a "No Data Columns Returned" message may result instead of an answer set containing the date values for which you are screening. For instance, the following WHERE clause would return '0' records for a YMD date value of 19000101 in a FOCUS database:

```
SELECT YMDFLD
FROM DATEFOC
WHERE YMDFLD = '00-01-01';
```

The following request, would return the desired value:

```
SELECT YMDFLD
FROM DATEFOC
WHERE YMDFLD = '1900-01-01';
```

SQL Scenario 6: Legacy Numeric Partial Date Container Columns - All Releases

If the date columns are numeric partial date containers stored in the database without an implicit century (these date columns might be described in an EDA Master File description using numeric legacy date formats such as I6YMD, I6MDY, I6DMY, I4YM, etc.), and are being retrieved from an EDA Hub or Full Function Server, then these fields will be converted by EDA to YYYYMMDD format upon output with '19' as the default century for all date values returned. For date types such as I4YM, which do not include a day component, a default value of the first day of the month will be assigned. The chart below illustrates this:

Date Value	EDA Master File Usage Format	Value Returned by EDA
001231	I6YMD	19001231
981125	I6YMD	19981125
123100	I6MDY	19001231
112598	I6MDY	19981125
311200	I6DMY	19001231
251198	I6DMY	19981125
0012	I4YM	19001201
9811	I4YM	19981101

What are the Year 2000 considerations for these types of columns?

Legacy date columns which do not store an explicit or implicit century are the highest risk date data types for potential Year 2000 problems. Even though EDA assigns 1900 as the default century when returning these columns, there may be situations where this is incorrect or inapplicable. Here's an example:

The following Master File description on an EDA Server describes a data file which stores expected graduation dates for students in a university:

```
FILE=STUDENT, SUFFIX=VSAM,
SEGNAME=ROOT, SEGTYPE=S0,
GROUP=STUDGRP, ALIAS=KEY, FORMAT=A9, ACTUAL=A9, $
FIELD=STUDENT_ID, ALIAS=, FORMAT=A9, ACTUAL=A9, $
FIELD=STUDENT_NAME, ALIAS=SNAME, FORMAT=A30, ACTUAL=A30, $
FIELD=GRAD_DATE, ALIAS=GDATE, FORMAT=I6YMD, ACTUAL=I6, $
FIELD=START_DATE, ALIAS=SDATE, FORMAT=I6YMD, ACTUAL=I6, $
FIELD=DATE_OF_BIRTH, ALIAS=DOB, FORMAT=I6YMD, ACTUAL=I6, $
```

EDA

Values in Student data file:

STUDENT_ID	STUDENT_NAME	GRAD_DATE	START_DATE	DATE_OF_BIRTH
123456789	Gary Johnson	990515	950906	751109
012345678	Nicole Younger	010628	970915	780612

From the EDA/Worksheet the following query is sent to the EDA Server:

```
SELECT GRAD_DATE, STUDENT_ID, STUDENT_NAME, START_DATE, DATE_OF_BIRTH  
FROM STUDENT  
ORDER BY GRAD_DATE;
```

The answer set returned by EDA:

GRAD_DATE	STUDENT_ID	STUDENT_NAME	START_DATE	DATE_OF_BIRTH
19010628	012345678	Nicole Younger	19950906	19751109
19990515	123456789	Gary Johnson	19970915	19780612

The information and sort order of the information in the report above is probably not what the user would have expected, since the output shows one student, Nicole Younger, is graduating in the year 1901, many years before she was born. She is also listed before student Gary Johnson, who is expected to graduate in the year 1999 (Nicole’s graduation date should be the year 2001).

Simply changing the USAGE format of these columns to a full-date container or Smart Date will not solve the problem, since there is no explicit or implicit century stored with the date value. EDA will simply assign these values a default century of ‘19.’ These situations offer the following options for remediation:

1. Leave the date value in the database unchanged, and assign a default century and threshold year that is more applicable to the application and the data. See the section, How to Remediate, for some examples on how this can be accomplished using the sliding date window.
2. Expand the date value in the database to include an explicit century. The sliding date window technique can also be used here. See the section entitled, How to Remediate, for some examples of how this can be accomplished.

Screening on these date fields is also a consideration. The default century value of ‘19’ must always be included when screening on these dates in an SQL SELECT statement, or you may receive either ‘0’ records or a “No Data Columns Returned” message instead of an answer set containing date values. Look at the following scenario:

The Master File description on EDA Server contains:

```
FILE=TESTDB, SUFFIX=FIX
SEGNAME=ONE
FIELD=SSN          , SSN, A9    , A9,$
FIELD=DATE_OF_BIRTH, DOB, I6YMD, I6,$
```

The Testdb database contains the following values:

```
SSN          DATE_OF_BIRTH
123456789    971227
```

From the EDA/Worksheet, the following SQL request is sent to the EDA Server:

```
SELECT SSN, DATE_OF_BIRTH
FROM TESTDB
WHERE DATE_OF_BIRTH = '97-12-27';
```

The result is that the message “No Data Columns Returned” is displayed, even though the screening condition essentially matches values in the database. To get the expected answer set returned, there are three options:

1. Include the default century ‘19’ in the screening condition. The query below will return the desired answer set:

```
SELECT SSN, DATE_OF_BIRTH
FROM TESTDB
WHERE DATE_OF_BIRTH = '1997-12-27';
```

2. Leave the date value in the database unchanged, and assign a more appropriate default century and threshold year. See the section entitled, How to Remediate, for some examples on how this can be accomplished using the sliding date window.
3. Expand the date value in the database to include an explicit century. The sliding date window technique can also be used here. See the section entitled, How to Remediate, for some examples on how this can be accomplished.

SQL Scenario 7: Legacy Alphanumeric Partial Date Container Columns - EDA 3.3.2 (MVS) & 4.x (All Platforms)

If the date columns are alphanumeric partial date containers that are stored in the database without an implicit century (these date columns might be described in an EDA Master File description using alphanumeric legacy date formats such as A6YMD, A6MDY, A6DMY, A4YM, etc.), and are being retrieved from an EDA Hub or Full Function Server, then these fields will be converted by EDA to YYYYMMDD format upon output. They will also be returned with ‘19’ as the default century for all date values returned. For date types such as A4YM, which do not include a day component, a default value of the first day of the month will be assigned. The chart below illustrates this:

Date Value	EDA Master File Usage Format	Value Returned by EDA
001231	A6YMD	19001231

EDA

981125	A6YMD	19981125
123100	A6MDY	19001231
112598	A6MDY	19981125
311200	A6DMY	19001231
251198	A6DMY	19981125
0012	A4YM	19001201
9811	A4YM	19981101

What are the Year 2000 considerations for these types of columns?

Legacy date columns which do not store an explicit or implicit century are the highest risk date data types for potential Year 2000 problems. Even though EDA assigns 1900 as the default century when returning these columns, there may be situations where this is incorrect or inapplicable. SQL Scenario 6 provides an example of when this situation could occur, along with other items to consider.

SQL Scenario 8: Legacy Alphanumeric Partial Date Container Columns - EDA 3.3.1 & Below

If the date columns are alphanumeric partial date containers that are stored in the database without an implicit century (these date columns might be described in an EDA Master File description using alphanumeric legacy dates such as A6YMD, A6MDY, A6DMY, A4YM, etc.), and are being retrieved from an EDA Hub or Full Function Server, then these columns will be returned in the same date component order as they were assigned in the EDA Master File description, and will not return any default century. The following chart illustrates this:

Date Value	EDA Master File Usage Format	Value Returned by EDA
001231	A6YMD	001231
981125	A6YMD	981125
123100	A6MDY	123100
112598	A6MDY	112598
311200	A6DMY	311200
251198	A6DMY	251198
0012	A4YM	0012
9811	A4YM	9811

Note: The date value returned by EDA might display with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

What are the Year 2000 considerations for these types of columns?

Alphanumeric legacy date columns which do not store an explicit or implicit century are also at risk for potential Year 2000 problems. Since no default century is included with these columns, the dates returned can be ambiguous. A value of 01/01/01 could be interpreted as 19010101, 01011901, 20010101 or 01012001. Changing the USAGE format of these columns to a six-digit integer legacy date, or a partial- or full-date container Smart Date will allow the dates to be returned with a default century. This will not solve the problem, however, since there is no explicit or implicit century stored with the date values. EDA will simply assign the default century of '19' to these values. For these situations, try the following options for remediation:

1. Leave the date value unchanged, and assign your own default century. See the section entitled, How to Remediate, below, for some examples of how this can be accomplished using the sliding date window. However, expanding the display of a six-position date to include the century might adversely affect output screens and reports.
2. Expand the date value in the database to include an explicit century. The sliding date window technique can also be used here. See the section entitled, How to Remediate, for some examples on how this can be accomplished. However, expanding the six-position date to include the century might adversely affect output screens, reports, metadata descriptions and programs that reference these columns.

For TABLE requests, the USAGE and ACTUAL formats in the Master File description on the EDA Server, and the manner in which the date values are stored in the database, will determine how the date values are displayed. The TABLE request received by the server could be one contained in a stored procedure on the EDA Server, or one that is in a –REMOTE or FOCUS Personal Agent request, but would be against a non-SUFFIX=EDA Master File description. Depending on the type and contents of the request, FOCUS services might be required. See the EDA Server manual for your platform for more information on FOCUS services. Look at the following scenarios to see how date formats will be returned for TABLE requests.

TABLE Scenario 1: Full Date Container Columns (Numeric & Alphanumeric) - All EDA Releases

If the date columns are full date containers, and have numeric or alphanumeric date formats (these might be fields described in an EDA Master File Description as I8YYMD, I8MDYY, I8DMYY, YYMD, DMYY, MDYY, A8YYMD, or A8MDYY), then the dates will be returned with the date components (month, day, year) in the same order as specified by the USAGE or FORMAT attributes for these fields in the EDA Master File description. The explicit century values contained in the database will be returned. Based on how these fields are described in the EDA Master File description, the chart below illustrates how these columns would be returned by EDA:

Date Value	EDA Master File Usage Format	Value Returned by EDA
20001231	I8YYMD	2000/12/31
19001231	I8YYMD	1900/12/31

EDA

12312000	I8MDYY	12/31/2000
12311900	I8MDYY	12/31/1900
31122000	I8DMYY	31/12/2000
31121900	I8DMYY	31/12/1900
20001231	A8YYMD	2000/12/31
19001231	A8YYMD	1900/12/31
12312000	A8MDYY	12/31/2000
12311900	A8MDYY	12/31/1900
31122000	A8DMYY	31/12/2000
31121900	A8DMYY	31/12/1900

What are the Year 2000 considerations for these types of columns?

Explicit century values are already available in the database for these types of columns. Therefore, as long as the work areas in your application that receive these columns, or the output screens and reports that display them, can already accommodate dates that include the century, no adjustments are necessary to your display environment to achieve Year 2000 compliance.

TABLE Scenario 2: Implicit Century Partial Date Container Columns - All EDA Releases

If the date columns are partial date containers that are stored in the database with an implicit century, and they are described to EDA using a Smart Date format such as YMD, MDY, DMY, YM, etc., these fields will be returned by EDA in the same date component order (month, day, year) as specified by the USAGE and FORMAT attributes in the server metadata description. However, TABLE requests against these columns will not, by default, return the implicit century that was stored when the file was loaded. FOCUS databases and files that can be described to EDA with ACTUAL=DATE and a Smart Date USAGE in the Master File description (i.e. HOLD files created from FOCUS files containing Smart Date columns), have the ability to store an implicit century. Even though the implicit century value is not returned, sorting, screening, and date math operations will still function properly on these fields since the date value is stored internally as an integer representing the number of days that have elapsed since the base date of December 31, 1900. YM format columns will be returned in year/month format with no default day value. The chart below illustrates this:

Date value	EDA Master File Usage Format	Value Returned by EDA
20001231	YMD	00/12/31
19001231	YMD	00/12/31

12312000	MDY	12/31/00
12311900	MDY	12/31/00
31122000	DMY	31/12/00
31121900	DMY	31/12/00
200012	YM	00/12

What are the Year 2000 considerations for these types of columns?

FOCUS databases and files that can be described to EDA in the Master File description with ACTUAL=DATE and Smart Date USAGE format (i.e. HOLD files created from FOCUS files containing Smart Date columns), have the ability to store an implicit century. However, if the data entry set-up for these columns only permits a user to enter the last two-digits of the year, then a default century value of '19' is probably stored in the database. This default value may not be correct for all the date values stored in the database. The implications of this situation, and some remediation strategies, are explored in the Level 3 - Database Considerations section.

Also, since by default the century is not returned by EDA for a TABLE request against a column in one of these formats, the dates returned can be ambiguous. A value of 010101 could be interpreted as 19010101, 01011901, 20010101 or 01012001. You may want to consider expanding the date to a full-date container Smart Date format in order to return the century. However, this type of change might adversely affect output screens and reports. An example of how to accomplish this is provided in the How to Remediate section.

TABLE Scenario 3: Legacy Partial Date Container Columns (Numeric & Alphanumeric) - All EDA Releases

If the date columns are partial date containers that are stored in the database without an implicit century, and they are described to EDA using numeric or alphanumeric legacy date formats, such as I6YMD, I6MDY, I6DMY, I4YM, A6YMD, A6MDY, or A4YM, then these fields will be returned by EDA in the same date component order (month, day, year) as specified by the USAGE or FORMAT attributes in the EDA Server Master File description. Columns using the I4YM or A4YM format will be returned in year/month format with no default day value. TABLE requests against these columns would also not be assigned a default century upon return, as would be the case with SQL requests. The chart below illustrates this:

Date Value	EDA Master File Usage Format	Value Returned by EDA
001231	I6YMD	001231
981125	I6YMD	981125
123100	I6MDY	123100
112598	I6MDY	112598
311200	I6DMY	311200

251198	I6DMY	251198
0012	I4YM	0012
9811	I4YM	9811
001231	A6YMD	001231
981125	A6YMD	981125
123100	A6MDY	123100
112598	A6MDY	112598
311200	A6DMY	311200
251198	A6DMY	251198
0012	A4YM	0012
9811	A4YM	9811

What are the Year 2000 considerations for these types of columns?

Legacy date columns which do not store an explicit or implicit century are the highest risk date data types for potential Year 2000 problems. Since no default century is included with these columns when they are returned by EDA, the dates returned can be ambiguous. A value of 010101 could be interpreted as 19010101, 01011901, 20010101 or 01012001. For non-FOCUS files, you could change the USAGE format of these columns to full-date container Smart Dates, which will allow the value to be returned with a default century of '19.' This may solve the problem in some instances, but is not a perfect solution. Some of the problems associated with this are noted in the "Year 2000 Considerations" section of "SQL Scenario 6, Legacy Numeric Partial Date Container Columns." For FOCUS databases, making this change to the USAGE format is not possible, as the USAGE in FOCUS databases also affects storage. In other words, your dates will be returned with unpredictable values. Some other remediation options are:

1. Leave the date value unchanged and assign your own default century . See the section entitled, How to Remediate, for some examples of how this can be accomplished using the sliding date window. However, expanding the display of the six-position date to include the century might adversely affect output screens and reports.
2. Expand the date value in the database to include an explicit century. The sliding date window technique can also be used here. See the section entitled, How to Remediate, for some examples on how this can be accomplished. However, expanding the six-position date to include the century might adversely affect output screens, reports, metadata descriptions and programs that reference these columns.

Data Processing Considerations:

Is my front-end application Year 2000 compliant?

If the front-end application will be processing data returned from the EDA Server, the application should also be examined for elements such as computations, variables, functions, joins to other local databases, screening conditions, subroutines or other procedures that might reference date columns, especially ones that may have been converted from a two-digit year format (YY) to a four-digit year format (YYYY). If possible, any remediation to the application should normally be handled by the application developer. However, the ability to alter the application may be limited if it is a packaged application written by a third-party or vendor. This topic is discussed in the section Determining Where to Remediate.

Other Considerations:

How will changing an application affect its users?

Keep in mind that any changes made to a front-end application might need to be redistributed to the user community. This may not be a great concern if all users are accessing the front-end application from a central source, such as a LAN network, where maintenance and access to the application can be centralized. However, if the application has been distributed to many users, then redistributing the changed application might be problem. In this instance, making changes to either the data or the metadata on the EDA Server might be a better option. The section, Determining Where to Remediate, explores this topic in greater detail.

Are the EDA/API, EDA/Client or EDA/ODBC Extender software Year 2000 capable?

Year 2000 capability is not an issue for either the EDA/API, EDA/Client or EDA/ODBC Extender as these products do not directly manipulate data. The EDA/Client software is used to connect to, and transfer data to and from, an EDA Server. The EDA/API and EDA/ODBC Extender provide function calls that facilitate communication between the application and the EDA/server. Year 2000 capability is only relevant for the client application and EDA/server, where data storage, display and manipulation can occur. An updated listing of Year 2000 capability criteria for Information Builders products is available on the Information Builders web site at www.ibi.com.

Level 2 - EDA Server Considerations

An EDA environment can consist of one or many client applications connecting to one or many EDA Servers and/or subservers. Each EDA Server or subserver that might potentially perform some type of processing on dates, whether in stored procedures, programs or external procedures, or that acts as a warehouse for the data being used in the environment, will need to be examined for potential Year 2000 problems. In general, we want to look for date fields in stored procedures and metadata stored on each server. Within each stored procedure, date fields might be found in subroutines, functions, DEFINES, COMPUTES, headings, sorting, JOINS, Dialogue Manager amper-variables, third party programs called by CALLPGM or CALLIMS, RDBMS stored procedures, or other areas. So, let us look at the two main areas of consideration, Stored Procedures and Metadata, to see how and where changes might need to be made.

Stored Procedure Considerations

Locating Server Procedures

Before you can examine what stored procedures might need to be reviewed for potential Year 2000 problems, you will first need to be able to locate them. Your EDA Server administrator can most likely help you identify important stored procedure directories and libraries, but in general, the following describes where stored procedures are normally stored on the most common EDA platforms:

MVS

The EDA stored procedure library is normally allocated to ddname EDARPC in the EDA Server JCL (for example, //EDARPC DD DSN=prefix.EDARPC.DATA.), but the library could also be dynamically allocated to ddname EDARPC in following places:

- The SERVICE= block of the server services file. This is the file allocated to ddname EDASERVE in the EDA Server JCL.
- The server global, or a user profile (for example, DYNAM ALLOC FILE EDARPC DA prefix.EDARPC.DATA). The server global profile is allocated to ddname EDASPROF in the file pointed to by ddname EDASERVE in the EDA Server JCL. The server user profile library is allocated to ddname EDAPROF in the file pointed to by ddname EDASERVE in the EDA Server JCL.

VM

On the EDA Server VM ID, the stored procedures are files with a file type of FOCEXEC. These files reside on disks that must be accessible at the time the EDA Server is started.

UNIX

The UNIX EDA Server will find stored procedures if they are in the bin (\$EDACONF/bin) or catalog (\$EDACONF/catalog) subdirectories of the directory where the EDA Server has been configured, or if they are located along the directory path specified by the environmental variable EDAPATH. The \$EDACONF/bin and \$EDACONF/catalog directories are part of the normal server search path. EDAPATH is an additional variable that can be exported in the server startup file or in a global- or user-level profile to specify additional working directories to be searched by the server. More information on these variables can be found in the EDA UNIX server manual.

NT

The NT EDA Server will find stored procedures if they are in the bin or catalog subdirectories of the directory where the EDA Server has been configured, or if they are located along the directory path specified by the environmental variable EDAPATH. The EDACONF bin and catalog subdirectories are part of the normal server search path. EDAPATH is an additional variable that can be set in the server configuration file, or in a global- or user-level profile, to specify additional working directories to be searched by the server. More information on these variables can be found in the EDA NT server manual.

Reviewing Stored Procedures

When reviewing stored procedures for dates, the following areas should be considered:

Dialogue Manager Date Variables

- In particular, date variables which do not contain an explicit century.

For example:

```
&YMD, &DMY, &MDY, &DATE
```

Note: The above date variables will not make use of the DEFCENT or YRTHRESH settings.

- If an explicit century is needed, you could use the following Dialogue Manager variables instead:

```
&DATEYYMD, &DATEDMY, &DATEMDYY
```

Note: The above variables are available in all EDA Release 3.1 or higher servers.

- Static dates in Dialogue Manager variables

For example:

```
-SET &TDATE='010198';  
-DEFAULTS &TDATE='010198'
```

These dates would need to be manually expanded to include the century.

For example:

```
-SET &TDATE='01011998';
-DEFAULTS &TDATE='01011998'
```

Date Subroutines

The following date-related subroutines will make use of the DEFCENT or YRTHRESH settings as of EDA Release 3.3.1:

```
AYMD, AYM, YM,
CHGDAT, GREGDT,
JULDAT, DAxxx, DTxxx
```

Functions There are three date-related functions that can be used to subtract one conventional date from another. They are: YMD, MDY and DMY. These functions are not really necessary if the dates you are attempting to subtract have been described with Smart Date formats in the Master File description. Smart Dates allow you to perform date arithmetic correctly, even across leap years. However, since you cannot apply Smart Date formats to Dialogue Manager amper variables, these functions are still of use. Starting in Release 3.3.1 of the EDA Server, these functions are DEFCENT and YRTHRESH aware. So, with DEFCENT=19, YRTHRESH=50, you can subtract 12/31/99 from 00/01/01 and get the correct answer.

Dates In Report Headings, Footings, Subheads or Subfoots

For example:

```
TABLE FILE ABC
HEADING CENTER
" Report Date: 12/31/90"
```

or

```
TABLE FILE ABC
HEADING CENTER
"Report Date: &DATE"
```

Might be remediated by using:

```
TABLE FILE ABC
HEADING CENTER
" Report Date: 12/31/1990"
```

or

```
TABLE FILE ABC
HEADING CENTER
"Report Date: &DATEMDYY"
```

Sorting (BY or ORDER BY) On Dates

Sorting on a two-digit year

Sorting becomes a problem when only a two-digit year is used with legacy date formats. This is especially true when date values from more than one century are involved. The sort order can become reversed. That is, the year value “00” will be sorted prior to the year value “99.” Some examples of where this situation can occur are provided in the section, Level 1: Application Considerations.

External Sorts

If the data is being passed to an external sort, the sort will not interpret DEFCEM/YRTHRESH commands.

COMPUTE/DEFINEs On Date Fields

In general, you would want to look for any date fields referenced in COMPUTE or DEFINEs to see if any adjustments are necessary to either formats, values, computations, etc., especially if changes have been made to either the metadata or the data columns they reference.

WHERE/IF Screening On Dates

You may need to include the century, even when screening on partial date container dates, if SQL SELECT statements are sent to the server. See the Level 1: Application Considerations section for some examples of when this is necessary.

TABLE requests and Dialogue Manager –IF statements that contain screening criteria against date columns which have been expanded to include the century in either data or metadata descriptions, may need to be evaluated, as they might fail if they do not include the century. This is especially true for date columns that use legacy date formats such as I6YMD and A6YMD.

Also consider:

- Programs called via CALLPGM or CALLIMS that might reference dates.
- JOINs and MATCHes on date columns.

A JOIN or MATCH of two or more files may not work as expected if changes to either the metadata or data for date columns are not consistent across the common date fields used.

Metadata Considerations

Locating Server Metadata

Before you can begin to examine what metadata (also known as Master File descriptions, or synonyms) might need to be reviewed for potential Year 2000 problems, you first need to be able to locate them. If the EDA Server is a relational gateway, then no EDA metadata descriptions are used, so there is no EDA metadata to examine. If the EDA Server is a communications gateway, once again, there will be no metadata descriptions to examine, because data access does not occur on the gateway machine. If the EDA Server is a HUB or Full Function server, there will be metadata descriptions to examine on one or more servers and/or subservers. Your EDA Server administrator will likely be able to help you identify important Master File description directories and libraries, but, in general, the following indicates where these files are normally stored across EDA platforms:

MVS

The EDA master file library is normally allocated to ddname EDAMFD or MASTER in the EDA Server JCL (for example, //EDAMFD DD DSN=prefix.EDAMFD.DATA.), but it could also be dynamically allocated to ddname EDAMFD or MASTER in either of the following areas:

- The SERVICE= block of the server services file (the file allocated to ddname EDASERVE in the EDA Server JCL).
- The server global or a user profile (for example, DYNAM ALLOC FILE EDAMFD DA prefix.EDAMFD.DATA). The server global profile is allocated to ddname EDASPROF in the file pointed to by ddname EDASERVE in the EDA Server JCL. The server user profile library is allocated to ddname EDAPROF in the file pointed to by ddname EDASERVE in the EDA Server JCL.

Synonym descriptions are stored in the destination pointed to by ddname EDASYNM, which could be allocated in the EDA Server JCL, the server services file or a global or user profile as indicated above.

VM

On the EDA Server VM ID, the metadata descriptions are files that contain the file type MASTER, and are on disks accessible at the time the EDA Server is started.

UNIX

Metadata descriptions will be found by the UNIX EDA Server if they are in the bin (\$EDACONF/bin) or catalog (\$EDACONF/catalog) subdirectories of the directory where the EDA Server has been configured, or if they are located along the directory path specified by the environmental variable EDAPATH. The \$EDACONF/bin and \$EDACONF/catalog directories are part of the normal server search path. EDAPATH is an additional variable that can be exported in the server startup file or in a global- or user-level profile to specify additional working directories to be searched by the server. More information on these variables can be found in the EDA UNIX server manual.

NT

Metadata descriptions will be found by the NT EDA Server if they are in the bin or catalog subdirectories of the directories where the EDA Server has been configured, or if they are located along the directory path specified by the environmental variable EDAPATH. The EDACONF bin and catalog subdirectories are part of the normal server search path. EDAPATH is an additional variable that can be set in the server configuration file, or in a global- or user-level profile to specify additional working directories to be searched by the server. More information on these variables can be found in the EDA NT server manual.

Reviewing EDA Metadata Descriptions for Dates

The information below can help you identify dates in EDA metadata descriptions. In general, if all dates in your metadata descriptions and data are using full-date containers, such as YYMD, then your metadata descriptions are Year 2000 compliant. However, in many cases storing the century value in EDA is not required, since EDA SQL requests return a default century. See the Level 1: Application Considerations section for more information on this topic. Finally, when reviewing EDA metadata descriptions for dates, consider legacy dates. In particular, consider the date variables which do not contain an explicit century. For example:

I4, I4YM, I6, I6YMD, I6MDY, I6DMY, A6, A6YMD, A6MDY, and A6DMY

Some of the options available for working with these date formats are explored in the section, Level 1: Application Considerations.

Level 3 - Database Considerations

Determining Whether to Remediate

To remediate or not to remediate. That is one of the questions you will probably be asking during this Year 2000 examination process. Another might be:

Should you reload the database to store the century explicitly?

Before you convert your entire database from two- to four-digit years, consider the impact this change may have on all the applications that use this file, and all of the metadata descriptions that may point to this file from EDA Servers and subservers. This may not necessarily be the best approach to making the application Year 2000 compliant. It may be enough to use a sliding date window, i.e. DEFCENT and YRTHRESH, to store and calculate the century.

Here are some factors to consider before making your decision:

- **Can the date values in my file be described within a 100-year window?**

If the existing date values in your file can be described within a 100-year range (for example, 1920 - 2019), then using the DEFCENT and YRTHRESH parameters may be an alternative to reloading the database to store an explicit century. If the existing date values do not fit into a 100-year span, then it may be necessary to reload the database.

- **If the file is a FOCUS database, is the implicit century stored with the Smart Date format columns (such as YMD, DMY, MDY, etc.) correct?**

A FOCUS database will store an implicit century with Smart Date format columns such as YMD, DMY or MDY when the database is loaded. If no century was included when the year was entered into the date column of the FOCUS database, the default century value '19' will be assigned to that year. In this instance, setting DEFCENT and YRTHRESH will not change the default century returned, as DEFCENT and YRTHRESH cannot change an existing century value. This holds true whether the century value was implicitly or explicitly provided in the database. In this instance, it may be necessary to reload the date values to include the desired century.

- **Can report widths or output screens be widened to accommodate date fields that have been expanded to include four-digit years?**

The report may already be at maximum width, and the output screens may not have the room to accommodate one or more expanded dates. In this case you may want to use a sliding date window, i.e. DEFCENT and YRTHRESH, to store and calculate the century.

- **Did you check the logic for all the applications that go against these date fields?**

Expanding the dates stored in the database might solve the Year 2000 problem for one application that accesses the database, but could also create unexpected problems for others that are designed to handle two-digit year values. Be careful to consider the “ripple effects” of any change to your database or your application.

- **Are these date fields used in any JOINS?**

If you change a data field in one file, you may also need to make corresponding changes in the target file.

- **How are these dates stored in historical databases?**

Determine whether you will still be able to reference, JOIN to, or screen on dates in historical databases if the data or metadata descriptions are changed.

- **How does this impact the screen for user input?**

Users may find it annoying to always enter a century in instances where the century value is redundant or is never in question. Also, determine whether there is room on the screen to add the space required to enter two additional bytes of data.

If it becomes necessary to change the data in the database, then it will be essential to locate the data. The EDA Server administrator and database administrator will probably be involved in this effort at some level, and should be consulted before any structural changes are made to the database.

Locating Server Data

Locating the data files available in your EDA Server environment is dependent upon the type of EDA Server that has been configured in your environment, as well as the type of file and platform on which the file resides. If the EDA Server is a relational gateway, the primary files returned from the EDA relational catalog will be those associated with the owner names listed in the EDA relational gateway SYSOWNER file. However, the primary files could also be any relational files a user can select against. See the EDA Server manual for your platform for more information on primary files.

A communications gateway server has no local data sources, since it only routes requests to other EDA Servers. An EDA Hub Server can have local data sources, but primarily points to, and describes data on, other EDA Servers. The SERVER= attribute of the Access File description for the SUFFIX=EDA Master File description on the Hub Server points to the EDA Server where the data resides. The TABLENAME attribute in the Access File description for the SUFFIX=EDA Master File description on the Hub Server indicates the name of the file on the EDA Server where the data resides. The data available to a Full Function EDA Server is usually local to the EDA Server. Below is a general guideline that can be used to help locate data files available to an EDA Full Function server or a Hub Server with local data access across platforms:

EDA

MVS

FOCUS, flat or VSAM data files may be allocated in the EDA Server JCL, or via the DYNAM command in the EDA service block, the global/user profile or in a stored procedure. The location and/or name of data files for relational and other non-relational data sources can be found in the Access File description. The Access File description is usually allocated to ddname EDAAFD or ACCESS in the EDA Server JCL, the service block, a global or user profile, or in a stored procedure.

VM

On the EDA Server VM ID, the data and Access File descriptions should be on disks accessible when the EDA Server is started.

UNIX

FOCUS files will be found in the configured server's binary or catalog directory, along the path specified by the EDAPATH environmental variable, or in the directory specified by a USE statement. A FILEDEF command is required for most non-relational files, and this could be set within a server global or user profile, or within a stored procedure. The Access File description for relational files will be found in the configured server's binary or catalog directory, or along the path specified by the EDAPATH environmental variable. The Access File has a file type of .ACX.

NT

FOCUS files will be found in the configured server's binary or catalog directory, along the path specified by the EDAPATH environmental variable, or in the directory specified by a USE statement. A FILEDEF command is required for most non-relational files, and this could be set within a server global or user profile, or within a stored procedure. The Access File description for relational files will be found in the configured server's binary or catalog directory, or along the path specified by the EDAPATH environmental variable. The Access File has a file type of .ACX.

How to Remediate

Now that we've explored the questions of where, what and whether to remediate, the next step is to determine how to remediate. In this section we'll take a more in-depth look at some of the options available.

Sliding Date Windows – DEFCENT and YRTHRESH

In Release 3.3.1 and higher EDA Servers, two SET commands provide a means of interpreting the century if the first two digits of the year are not provided. These SET commands are DEFCENT and YRTHRESH. The combination of DEFCENT and YRTHRESH establishes a base year and a 100-year window for determining the century value for a two digit year. When a date value has only two digits for the year, the century will be determined by fitting these two digits within the declared window. Any two-digit years that are greater than or equal to the threshold year (YRTHRESH) value will assume the century value assigned by DEFCENT. Any two-digit year values that are lower than that threshold (YRTHRESH) value will assume the century value of DEFCENT plus one. Years outside the declared window must be handled by user coding. The default values for the two commands are SET DEFCENT= 19, SET YRTHRESH= 00.

The syntax is:

```
SET DEFCENT = nn 19
SET YRTHRESH= nn 00
```

(**Note:** The underscored values are the defaults.)

Here are some examples of the settings and the 100-year span that is represented by each pair of settings:

DEFCENT	YRTHRESH	100-year span	Two-digit year supplied	Resulting four-digit year
19	00	1900-1999	40	1940
			90	1990
19	50	1950-2049	40	2040
			90	1990

In the first example, the default values of DEFCENT and YRTHRESH describe a range from 1900 to 1999. If a 2-digit year field contains the value 90, then EDA interprets the year as 1990. If the year field is 40, then the year is interpreted as 1940. If the year field is 00, then the year is interpreted as 1900. However, in the second example, if DEFCENT is set to 19 and YRTHRESH to 50, then the new 100-year range is 1950 through 2049. The century value for any 2-digit year value of 49 or under would be interpreted as 20, and for any 2-digit year values from 50 through 99, the century value would be interpreted as 19.

Setting DEFCENT and YRTHRESH

It is possible to set DEFCENT and YRTHRESH at five levels within the EDA Server environment. The levels are:

1. In the EDA Server Global or User Profile

On the EDA Server, add the following SET commands into either the EDA Server global or user profile, or into a stored procedure:

```
SET DEFCENT = nn
```

where:

nn = 19 unless otherwise specified.

```
SET YRTHRESH = nn
```

where:

nn = 0 unless otherwise specified.

The current settings of DEFCENT and YRTHRESH for your EDA Server connected session can be viewed by executing the EDAEXEC stored procedure and providing “? SET” as a parameter. For example, from an RDAAPP front- end issue:

```
X EDAEXEC ? SET
```

Note: If these settings are included in the global profile, then all users who connect to that EDA Server will have these settings available. If they are set in the user profile or in a stored procedure, these settings will only be available to specified users or whoever executes the stored procedure. Setting these parameters at the environmental level will affect all requests containing partial date containers that meet the criteria specified below, in the section entitled, When Will DEFCENT and YRTHRESH Be Used?

2. At the File Level in an EDA Master File Description

These settings can be added at the file level in an EDA Master File description (synonym). The settings FDEFCENT and FYRTHRESH are used at the file level. The FDEFCENT syntax is

```
{FDEFCENT }  
{FDFC      }= nn
```

where:

nn = 19, unless otherwise specified.

The FYRTHRESH syntax is

```
{FYRTHRESH }  
{FYRT      }= nn
```

where:

nn = 0, unless otherwise specified.

Example:

```
FILE=EDUCFILE, SUFFIX=FOC, FDEFCENT=19, FYRTHRESH=50, $
```

Note: Providing these settings at the file level will only affect partial date container columns within this file that meet the criteria specified below, in the section entitled, When Will DEFCENT and YRTHRESH Be Used?

3. At the Field Level in an EDA Master File Description

On the EDA Server, the DEFCENT and YRTHRESH settings can be added at the field level in an EDA Master File description (synonym). The syntax is

```
{DEFCENT }
{DFC      }= nn
```

where:

nn = 19, unless otherwise specified.

and

```
{YRTHRESH }
{YRT      }= nn
```

where:

nn = 0, unless otherwise specified.

Example:

```
FILE = ALUMNI, SUFFIX = FIX, $
SEGNAME = SEG1, SEGTYPE = S1
FIELD = ALUMI_NO ,ALIAS = SSN      , USAGE = A9 , ACTUAL = A09,$
FIELD = NAME , ALIAS = NAME , USAGE = A20 , ACTUAL = A20,$FIELD =
DT_OF_BIRTH , ALIAS = DOB , USAGE = I6YMD, ACTUAL = A06,
      DEFCENT = 20, YRTHRESH =04,$
$Year Span 2004 → 2103
```

Note: Providing these settings at the field level will only affect the field to which these settings are being applied.

4. In a DEFINE Field in the EDA Master File Description

On the EDA Server, these settings can be added in a DEFINE field in the EDA Master File description.

The syntax for the DEFINE field in the master file is:

```
DEFINE fld/fmt {DEFCENT} {YRTHRESH}
               {DFC      } {YRT      } [MISSING...] = exp;$
```

Note: The DEFCENT and YRTHRESH syntax must follow the field format information for the DEFINED field.

Example:

EDA

```
FILE = ALUMNI, SUFFIX = FIX, $
SEGMNAME = SEG1, SEGTYPE = S1
FIELD = ALUMI_NO , ALIAS = SSN , USAGE = A9 , ACTUAL = A09,$
FIELD = NAME , ALIAS = NAME , USAGE = A20 , ACTUAL = A20,$
FIELD = DT_OF_BIRTH , ALIAS = DOB , USAGE = A6YMD, ACTUAL = A06,$

$Year Span 2004 → 2103
DEFINE NEW_DOBFLD/YYMD DEFCENT 20 YRTHRESH 04 =DT_OF_BIRTH;$
```

Note: Applying these settings to a DEFINED field at the field level will only affect that DEFINED field.

5. On the EDA Server, Within a Stored Procedure

On the EDA Server, within a stored procedure, these settings can be added in a DEFINE statement preceding a TABLE request that utilizes the date columns.

The syntax for the DEFINE in a stored procedure is:

```
DEFINE FILE filename
      {DEFCENT} {YRTHRESH}
fld/fmt {DFC } {YRT } [MISSING...] = exp;$
END
```

Note: The DEFCENT and YRTHRESH syntax must follow the field format information for the defined field information.

Example:

```
DEFINE FILE ALUMNI
FULL_DOB/YYMD DEFCENT 20 YRTHRESH 04 =DT_OF_BIRTH;
END
TABLE FILE ALUMNI
PRINT FULL_DOB
END
```

Note: Applying these settings to a DEFINED field at the field level will only affect that DEFINED field.

Order of Execution, By DEFINE Level

If DEFCENT and YRTHRESH were set at all of the levels indicated above, the execution order would be as follows:

1. Field Level

This includes the possibility that DEFCENT and YRTHRESH were set as a DEFINE in the EDA Master File description or in a DEFINE in a stored procedure.

2. File Level

3. Environment Level

If DEFCENT and YRTHRESH are set at the field level, any settings at the file level will be overridden. If they are set at the file level, any settings at the global (environment) level will be overridden. Settings at the global (environment) level apply if DEFCENT and YRTHRESH are not being set anywhere else.

When Will DEFCENT and YRTHRESH Be Used?

The DEFCENT and YRTHRESH settings will be used when date conversion takes place. For example:

- Converting legacy dates to Smart Dates when the first two digits are not supplied (for example, I6YMD -> YYMD).
- Converting from ACTUAL to USAGE when the first two digits are not supplied (for example, USAGE=YYMD,ACTUAL=I6).
- Data entry (via FIXFORM), when the first two digits are not provided (for FOCUS databases).

Let's take a look at how DEFCENT and YRTHRESH will affect the century value assigned to a date and solve some of the SQL and TABLE scenario problems presented in the *Applications Considerations* section.

SQL Scenario 1, 2, 3 and 4: The date columns are DATE data type columns in a relational table and they are being retrieved from an EDA relational gateway server or in SQL passthru mode

or

The date columns are full date containers with numeric or alphanumeric date formats (YYMD, I8YYMD, I8MDYY, I8DMYY, A8YYMD, A8MDYY, A8DMYY).

Setting DEFCENT and YRTHRESH will have no effect if the century is explicitly supplied. If a century is explicitly supplied in the database, it will be returned.

SQL Scenario 5:

The date columns are partial date containers that are stored in the database with an implicit century, and they are described to EDA using Smart Date formats (for example, YMD, MDY, DMY, YM, etc.).

Setting DEFCENT and YRTHRESH will have no effect if the century is implicitly supplied. If a century is implicitly supplied in the database, it will be returned.

SQL Scenario 6 and 7:

The date columns are partial date containers that are stored in the database without an implicit century, and they are described to EDA using numeric legacy date format (for example, I6YMD, I6MDY, I6DMY, I4YM, etc.).

or

The date columns are partial date containers that are stored in the database without an implicit century, and they are described to EDA using alphanumeric legacy date format (i.e. A6YMD, A6MDY, A6DMY, A4YM, etc.), and retrieval is against an EDA Release 3.3.2 (MVS) or 4.x (all platforms) server.

In these scenarios, DEFCENT and YRTHRESH will be applied, and will affect the century returned in the answer set returned from the EDA Server. What follows is the sample output from an SQL query both before and after DEFCENT and YRTHRESH are set to values other than their defaults.

SQL Request:

```
SELECT I6YMDFLD, I6MDYFLD, I4YMFLD,
       A6YMDFLD, A6MDYFLD, A4YMFLD
FROM DATEFILE;
```

Master File description before changes:

```
FILE=DATEFILE          ,SUFFIX=FIX
SEGNAME=DATEFILE,SEGTYPE=S0
FIELDNAME  =YYMDFLD      ,E01    ,YYMD    ,A08    , $
FIELDNAME  =I6YMDFLD    ,E02    ,I6YMD   ,A06    , $
FIELDNAME  =I6MDYFLD    ,E03    ,I6MDY   ,A06    , $
FIELDNAME  =I4YMFLD     ,E04    ,I4YM    ,A04    , $
FIELDNAME  =A6YMDFLD    ,E02    ,A6YMD   ,A06    , $
FIELDNAME  =A6MDYFLD    ,E03    ,A6MDY   ,A06    , $
FIELDNAME  =A4YMFLD     ,E04    ,A4YM    ,A04    , $
```

Output before Master File description changes, with DEFCENT and YRTHRESH at the default values:

Date Value	Report Column	Value Returned by EDA
001231	I6YMDFLD	19001231
981125	I6YMDFLD	19981125
123100	I6MDYFLD	19001231
112598	I6MDYFLD	19981125
311200	I6DMYFLD	19001231
251198	I6DMYFLD	19981125
0012	I4YMFLD	19001201

9811	I4YMFLD	19981101
001231	A6YMDFLD	19001231
981125	A6YMDFLD	19981125
123100	A6MDYFLD	19001231
112598	A6MDYFLD	19981125
311200	A6DMYFLD	19001231
251198	A6DMYFLD	19981125
0012	A4YMFLD	19001201
9811	A4YMFLD	19981101

Master File description after changes:

```

FILE=DATEFILE          , SUFFIX=FIX
SEGNAME=DATEFILE , SEGTYPE=S0
FIELDNAME  =YYMDFLD      ,E01  ,YYMD  ,A08      , $
FIELDNAME  =I6YMDFLD     ,E02  ,I6YMD ,A06 ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =I6MDYFLD     ,E03  ,I6MDY ,A06 ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =I4YMFLD      ,E04  ,I4YM  ,A04 ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A6YMDFLD     ,E02  ,A6YMD ,A06 ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A6MDYFLD     ,E03  ,A6MDY ,A06 ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A4YMFLD      ,E04  ,A4YM  ,A04 ,
  DEFCENT=19, YRTHRESH=50,$
    
```

Output after Master File description changes, with DEFCENT=19 and YRTHRESH=50:

Date Value	Report Column	Value Returned by EDA
001231	I6YMDFLD	20001231
981125	I6YMDFLD	19981125
123100	I6MDYFLD	20001231
112598	I6MDYFLD	19981125
311200	I6DMYFLD	20001231
251198	I6DMYFLD	19981125
0012	I4YMFLD	20001201
9811	I4YMFLD	19981101

EDA

001231	A6YMDFLD	20001231
981125	A6YMDFLD	19981125
123100	A6MDYFLD	20001231
112598	A6MDYFLD	19981125
311200	A6DMYFLD	20001231
251198	A6DMYFLD	19981125
0012	A4YMFLD	20001201
9811	A4YMFLD	19981101

Note: The date values returned by EDA might display with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

SQL Scenario 8:

The date columns are partial date containers that are stored in the database without an implicit century, and they are described to EDA using alphanumeric legacy date formats (for example, A6YMD, A6MDY, A6DMY, A4YM, etc.).

If the EDA Release level is 3.3.1, then DEFCENT and YRTHRESH will be applied. However, since an SQL request against an alphanumeric date will not automatically return the century in EDA Server Releases 3.3.1 and below, you can allow the century to be returned in one of the four ways described here.

Note: All four methods require an EDA Release 3.3.1 level server or higher for support of the DEFCENT/YRTHRESH parameters.

1. If the file is a non-FOCUS database, change the USAGE attribute in the Master File description from the A6 date format to a full date container Smart Date format (for example, YYMD). Likewise, if the USAGE attribute is an A4 date format, change it to YYM. Then apply DEFCENT and YRTHRESH at the field level in the Master File description. Below are examples of the Master File description and output from an SQL request before and after making the changes specified above:

SQL request:

```
SELECT A6YMDFLD, A6MDYFLD, A4YMFLD
FROM DATETEST;
```

Master file before changes:

```
FILE=DATETEST, SUFFIX=FIX,
SEGNAME=DATETEST, SEGTYPE=S0,
FIELD=A6YMDFLD, ALIAS=E01, USAGE=A6YMD,ACTUAL=A06,$
FIELD=A6MDYFLD, ALIAS=E02, USAGE=A6MDY,ACTUAL=A06,$
FIELD=A4YMFLD , ALIAS=E03, USAGE=A4YM,ACTUAL=A04,$
```

Output before changes (with default DEFCENT and YRTHRESH values):

Date Value	EDA Master File Column	Value Returned by EDA
001231	A6YMDFLD	001231
981125	A6YMDFLD	981125
123100	A6MDYFLD	123100
112598	A6MDYFLD	112598
0012	A4YMFLD	0012
9811	A4YMFLD	9811

Note: The date value returned by EDA might display with or without slashes or dashes between the year, month and day, depending on the EDA front-end application used.

Master File description after changes:

```
FILE=DATETEST          ,SUFFIX=FIX
SEGNAME=DATETEST  ,SEGTYPE=S0
FIELDNAME   =A6YMDFLD          ,E01          ,YYMD          ,A06          ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME   =A6MDYFLD          ,E02          ,MDYY          ,A06          ,
  DEFCENT=19, YRTHRESH=50,$
FIELDNAME   =A4YMFLD           ,E03          ,YYM           ,A04          ,
  DEFCENT=19, YRTHRESH=50,$
```

Output after changes:

Date Value	EDA Master File Column	Value Returned by EDA
001231	A6YMDFLD	20001231
981125	A6YMDFLD	19981125
123100	A6MDYFLD	20001231
112598	A6MDYFLD	19981125
0012	A4YMFLD	20001201
9811	A4YMFLD	19981101

- Another way that DEFCENT/YRTHRESH can be applied to make an alphanumeric date in a NON-FOCUS file available for SQL queries against the file is to add DEFINED fields to the Master File description as follows:

EDA

```
FILE=DATETEST, SUFFIX=FIX,
SEGNAME=DATETEST, SEGTYPE=S0,
FIELD=A6YMDFLD, ALIAS=E01, USAGE=A6YMD,ACTUAL=A06,$
FIELD=A6MDYFLD, ALIAS=E02, USAGE=A6MDY,ACTUAL=A06,$
FIELD=A4YMFLD , ALIAS=E03, USAGE=A4YM,ACTUAL=A04,$
DEFINE DA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=A6YMDFLD;
DEFINE DA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=A6MDYFLD;
DEFINE DA4YMFLD/YM DEFCENT 19 YRTHRESH 50=A4YMFLD;
```

The SQL request would then reference the DEFINEd fields in the query:

```
SELECT DA6YMDFLD, DA6MDYFLD, DA4YMFLD
FROM DATETEST;
```

3. If the file is a FOCUS database, then changing the USAGE attribute in the Master File description from the A6 date format to a full date container Smart Date (for example, YYMD) would yield a corrupted date value. This happens because the USAGE attribute for a FOCUS file affects how the value is stored in the database. The way to apply DEFCENT and YRTHRESH so that the settings are available for SQL queries against the file is by adding DEFINEd fields to the Master File description as follows:

```
FILE=DATEFOC, SUFFIX=FOC,
SEGNAME=DATEFOC, SEGTYPE=S1,
FIELD=A6YMDFLD, ALIAS=E01, USAGE=A6YMD,$
FIELD=A6MDYFLD, ALIAS=E02, USAGE=A6MDY,$
FIELD=A4YMFLD , ALIAS=E03, USAGE=A4YM,$
DEFINE DA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=A6YMDFLD;$
DEFINE DA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=A6MDYFLD;$
DEFINE DA4YMFLD/YM DEFCENT 19 YRTHRESH 50=A4YMFLD;$
```

The SQL request would then reference the DEFINEd fields in the query:

```
SELECT DA6YMDFLD, DA6MDYFLD, DA4YMFLD
FROM DATEFOC;
```

4. If upgrading the EDA Server is an option for your site, then an upgrade to EDA 3.3.2 (if the environment is MVS) or 4.x (for all platforms) servers, will provide the automatic functionality of having alphanumeric date fields returned with a default century. The techniques applied in SQL Scenario 7 above, can then also be used to assign your own default century using the sliding date window.

TABLE Scenario 1:

The date columns are full date containers, and they have numeric or alphanumeric date formats (for example, I8YYMD, I8MDYY, I8DMYY, YYMD, A8YYMD, A8MDYY).

Setting DEFCENT and YRTHRESH will have no effect if the century is explicitly supplied. If a century is explicitly supplied in the database, it will be returned.

TABLE Scenario 2:

The date columns are partial date containers that are stored in the database with an implicit century, and they are described to EDA using Smart Date formats (for example, YMD, MDY, DMY, YM, etc.).

Setting DEFCENT and YRTHRESH will have no effect if the century is implicitly supplied. If a century is implicitly supplied in the database, it will be returned. However, for a TABLE request, in order to see the implicit century stored with the date column, you would need to manually expand the column to a full-date container format. This would be done either in the Master File description or in a DEFINE. Since a Smart Date format is being used, you can easily accomplish this by directly changing the USAGE or FORMAT value from a partial date container Smart Date format (YMD, MDY or DMY) to a full-date container Smart Date format (YYMD, MDYY or DMYY) in the Master File description without affecting the storage of the values in the FOCUS file. This is because a Smart Date format column in a FOCUS file stores a date value as an integer representing the number of days that have elapsed since the base date of December 31, 1900. Because the value is being stored as the integer representation of the date instead of the static date value, it can be represented accurately with any Smart Date format. The example below shows the Master File description and output before and after making this change:

TABLE Request:

```
TABLE FILE DATEFOC
PRINT YMDFLD MDYFLD YMFLD
END
```

Master File before changes:

```
FILE=DATEFOC, SUFFIX=FOC,
SEGNAME=DATEFOC, SEGTYPE=S1,
FIELD=YMDFLD, ALIAS=E01, USAGE=YMD,$
FIELD=MDYFLD, ALIAS=E01, USAGE=MDY,$
FIELD=YMFLD, ALIAS=E01, USAGE=YM,$
```

Output before changes:

Date Value	Master File Column	Value Returned by EDA
20001231	YMDFLD	00/12/31
19001231	YMDFLD	00/12/31
12312000	MDYFLD	12/31/00
12311900	MDYFLD	12/31/00
190012	YMFLD	00/12
200012	YMFLD	00/12

Master File after changes:

EDA

```
FILE=DATEFOC, SUFFIX=FOC,
SEGNAME=DATEFOC, SEGTYPE=S1,
FIELD=YMDFLD, ALIAS=E01, USAGE=YYMD,$
FIELD=MDYFLD, ALIAS=E01, USAGE=MDYY,$
FIELD=YMFLD, ALIAS=E01, USAGE=YYM,$
```

Output after changes:

Date Value	Master File Column	Value Returned by EDA
20001231	YMDFLD	2000/12/31
19001231	YMDFLD	1900/12/31
12312000	MDYFLD	12/31/2000
12311900	MDYFLD	12/31/1900
190012	YMFLD	1900/12
200012	YMFLD	2000/12

TABLE Scenario 3:

The date columns are partial date containers that are stored in the database without an implicit century, and they are described to EDA using numeric or alphanumeric legacy date formats (for example, I6YMD, I6MDY, I6DMY, I4YM, A6YMD, A6MDY, A4YM, etc.).

In this case, DEFCENT and YRTHRESH will be applied, and they will determine the century returned in the answer set returned from the EDA Server. How the century is returned depends on how your situation fits into one of the following three categories:

1. If the file is a non-FOCUS database, change the USAGE attribute in the Master File description from the I6 or A6 date format to a full date container Smart Date (YYMD). Change an A4 or I4 date format to YYM and apply DEFCENT and YRTHRESH at the field level in the Master File description. Below is an example of the Master File description and output from an SQL request before and after making the changes specified above:

TABLE Request:

```
TABLE FILE DATEFILE
PRINT I6YMDFLD I6MDYFLD I4YMFLD
      A6YMDFLD A6MDYFLD A4YMFLD
```

END

Master File before changes:

How to Remediate

```
FILE=DATEFILE          ,SUFFIX=FIX
SEGNAME=DATEFILE ,SEGTYPE=S0
FIELDNAME  =I6YMDFLD      ,E02          ,I6YMD   ,A06     , $
FIELDNAME  =I6MDYFLD      ,E03          ,I6MDY   ,A06     , $
FIELDNAME  =I4YMFLD       ,E04          ,I4YM    ,A04     , $
FIELDNAME  =A6YMDFLD      ,E02          ,A6YMD   ,A06     , $
FIELDNAME  =A6MDYFLD      ,E03          ,A6MDY   ,A06     , $
FIELDNAME  =A4YMFLD       ,E04          ,A4YM    ,A04     , $
```

Output before changes, with DEFCENT and YRTHRESH at the default values.

Date Value	Date Column in Master File	Value Returned by EDA
001231	I6YMDFLD	00/12/31
981125	I6YMDFLD	98/11/25
123100	I6MDYFLD	12/31/00
112598	I6MDYFLD	11/25/98
0012	I4YMFLD	00/12
9811	I4YMFLD	98/11
001231	A6YMDFLD	00/12/31
981125	A6YMDFLD	98/11/25
123100	A6MDYFLD	12/31/00
112598	A6MDYFLD	11/25/98
0012	A4YMFLD	00/12
9811	A4YMFLD	98/11

Master File after changes:

```
FILE=DATEFILE          ,SUFFIX=FIX
SEGNAME=DATEFILE ,SEGTYPE=S0
FIELDNAME  =I6YMDFLD      ,E02          ,YYMD   ,A06     ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =I6MDYFLD      ,E03          ,MDY   ,A06     ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =I4YMFLD       ,E04          ,YY    ,A04     ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A6YMDFLD      ,E01          ,YYMD   ,A06     ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A6MDYFLD      ,E02          ,MDY   ,A06     ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A4YMFLD       ,E03          ,YY    ,A04     ,
DEFCENT=19, YRTHRESH=50,$
```

Output after changes, with DEFCENT=19 and YRTHRESH=50.

Date Value	Date Column in Master File	Value Returned by EDA
001231	I6YMDFLD	2000/12/31
981125	I6YMDFLD	1998/11/25
123100	I6MDYFLD	12/31/2000
112598	I6MDYFLD	11/25/1998
0012	I4YMFLD	2000/12
9811	I4YMFLD	1998/11
001231	A6YMDFLD	2000/12/31
981125	A6YMDFLD	1998/11/25
123100	A6MDYFLD	12/31/2000
112598	A6MDYFLD	11/25/1998
0012	A4YMFLD	2000/12
9811	A4YMFLD	1998/11

2. DEFCEMENT and YRTHRESH could also have been set in a DEFINE within the server Master File description or within a stored procedure that precedes a TABLE request.

Setting DEFCEMENT and YRTHRESH in a DEFINE within the server Master File description:

```

FILE=DATETEST, SUFFIX=FIX,
SEGNAME=DATETEST, SEGTYPE=S0,
FILE=DATEFILE, SUFFIX=FIX
SEGNAME=DATEFILE, SEGTYPE=S0
FIELDNAME =I6YMDFLD, E02, I6YMD, A06, $
FIELDNAME =I6MDYFLD, E03, I6MDY, A06, $
FIELDNAME =I4YMFLD, E04, I4YM, A04, $
FIELDNAME =A6YMDFLD, E02, A6YMD, A06, $
FIELDNAME =A6MDYFLD, E03, A6MDY, A06, $
FIELDNAME =A4YMFLD, E04, A4YM, A04, $
DEFINE IA6YMDFLD/YYMD DEFCEMENT 19 YRTHRESH 50=I6YMDFLD;$
DEFINE IA6MDYFLD/MDYY DEFCEMENT 19 YRTHRESH 50=I6MDYFLD;$
DEFINE IA4YMFLD/YYM DEFCEMENT 19 YRTHRESH 50=I4YMFLD;$
DEFINE DA6YMDFLD/YYMD DEFCEMENT 19 YRTHRESH 50=A6YMDFLD;$
DEFINE DA6MDYFLD/MDYY DEFCEMENT 19 YRTHRESH 50=A6MDYFLD;$
DEFINE DA4YMFLD/YYM DEFCEMENT 19 YRTHRESH 50=A4YMFLD;$
    
```

Alternatively, DEFCEMENT and YRTHRESH could have been set in a DEFINE in a stored procedure:

```

DEFINE FILE DATETEST
IA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=I6YMDFLD;
IA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=I6MDYFLD;
IA4YMFLD/YYM DEFCENT 19 YRTHRESH 50=I4YMFLD;
DA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=A6YMDFLD;
DA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=A6MDYFLD;
DA4YMFLD/YYM DEFCENT 19 YRTHRESH 50=A4YMFLD;
END

```

The TABLE request would then reference the DEFINED fields in the query:

```

TABLE FILE DATETEST
PRINT IA6YMDFLD IA6MDYFLD IA4YMFLD
      DA6YMDFLD DA6MDYFLD DA4YMFLD
END

```

3. If the file is a FOCUS database, then changing the USAGE attribute in the Master File description from the I6 or A6 date format to a full date container Smart Date (YYMD) would yield a corrupted date value. This happens because the USAGE attribute for a FOCUS file affects how the value is stored in the database. The way to apply DEFCENT and YRTHRESH so that they are available for SQL queries against the file is to add DEFINED fields to the Master File description. For example:

```

FILE=DATEFOC, SUFFIX=FOC,
SEGNAME=DATEFOC, SEGTYPE=S1,
FIELD=I6YMDFLD, ALIAS=E01, USAGE=I6YMD,$
FIELD=I6MDYFLD, ALIAS=E02, USAGE=I6MDY,$
FIELD=I4YMFLD , ALIAS=E03, USAGE=I4YM,$
FIELD=A6YMDFLD, ALIAS=E01, USAGE=A6YMD,$
FIELD=A6MDYFLD, ALIAS=E02, USAGE=A6MDY,$
FIELD=A4YMFLD , ALIAS=E03, USAGE=A4YM,$
DEFINE IA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=I6YMDFLD;$
DEFINE IA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=I6MDYFLD;$
DEFINE IA4YMFLD/YYM DEFCENT 19 YRTHRESH 50=I4YMFLD;$
DEFINE DA6YMDFLD/YYMD DEFCENT 19 YRTHRESH 50=A6YMDFLD;$
DEFINE DA6MDYFLD/MDYY DEFCENT 19 YRTHRESH 50=A6MDYFLD;$
DEFINE DA4YMFLD/YYM DEFCENT 19 YRTHRESH 50=A4YMFLD;

```

The TABLE request references the DEFINED fields in the query as follows:

```

TABLE FILE DATEFOC
PRINT IA6YMDFLD IA6MDYFLD IA4YMFLD
      DA6YMDFLD DA6MDYFLD DA4YMFLD
END

```

Creating a Century Aware Extract File for Files With Dates That Do Not Include the Century

The following technique uses DEFCENT and YRTHRESH parameters to create an extract file that can be used to reload a database with century-aware dates. You will need the following:

- EDA Server Release 3.3.1 or higher. Only these releases support DEFCENT and YRTHRESH.

EDA

- Write access to, and the authority to change, EDA metadata descriptions or data.

Before making any changes, be aware that you should:

- Make a backup copy of the database and Master File description.
- Test any changes on a backup or test copy of the database, and run that test copy against all applications before moving your changes to the production environment.

Note: The steps provided below have been tested in the MVS EDA environment only. Examples for other platforms will be provided in a future version of this document.

Step 1.

Make a copy of the Master File description.

Step 2.

Add DEFCENT and YRTHRESH to the copy of the Master File description using a method appropriate for your environment, based on the SQL and TABLE scenarios above. Choose a value of DEFCENT and YRTHRESH that suits your environment or data. We'll use the following sample Master File description and an SQL scenario:

Master File **before** changes:

```
FILE=DATEFILE          ,SUFFIX=FIX
SEGNAME=DATEFILE,SEGTYPE=S0
FIELDNAME  =I6YMDFLD          ,E02          ,I6YMD          ,A06          ,$
FIELDNAME  =I6MDYFLD          ,E03          ,I6MDY          ,A06          ,$
FIELDNAME  =I4YMFLD           ,E04          ,I4YM           ,A04          ,$
FIELDNAME  =A6YMDFLD          ,E02          ,A6YMD          ,A06          ,$
FIELDNAME  =A6MDYFLD          ,E03          ,A6MDY          ,A06          ,$
FIELDNAME  =A4YMFLD           ,E04          ,A4YM           ,A04          ,$
```

Copy of Master File description **including** changes:

```
FILE=DATECOPY ,SUFFIX=FIX
SEGNAME= DATECOPY, SEGTYPE=S0
FIELDNAME  =I6YMDFLD          ,E02          ,I6YMD          ,A06          ,
DEFCENT=19, YRTHRESH=50, $
FIELDNAME  =I6MDYFLD          ,E03          ,I6MDY          ,A06          ,
DEFCENT=19, YRTHRESH=50, $
FIELDNAME  =I4YMFLD           ,E04          ,I4YM           ,A04          ,
DEFCENT=19, YRTHRESH=50, $
FIELDNAME  =A6YMDFLD          ,E01          ,YYMD          ,A06          ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A6MDYFLD          ,E02          ,MDYY          ,A06          ,
DEFCENT=19, YRTHRESH=50,$
FIELDNAME  =A4YMFLD           ,E03          ,YYM          ,A04          ,
DEFCENT=19, YRTHRESH=50,$
```

Step 3.

Allocate space for a temporary file (the space allocation should be slightly larger than the original database). This space is needed for the extract file created from the original database. On MVS, the DCB parameters (LRECL, RECFM, and BLKSIZE) used to allocate this file will be irrelevant, as they will be overwritten when the file is created by EDA. Within the stored procedure created in Step 5, you will need to make sure there is an allocation pointing to the temporary file for which space is being allocated here. For example:

```
-*MVS
DYNAM ALLOC FILE DATETEMP DA prefix.DATETEMP.DATA SHR REU
```

Note: Refer to the EDA MVS server manual for more information on the DYNAM command.

Step 4.

Within the stored procedure you will create in Step 6 below, you will need to include an allocation that points the name (or ddname) of the copy of the Master File description created in Step 2, to the original database. This will enable the date values in the database to be extracted according to the new Master File description that accommodates the century values. Here's an example:

```
DYNAM ALLOC FILE DATECOPY DA prefix.DATEFILE.DATA SHR REU
```

Note: Refer to the EDA MVS server manual for more information on the DYNAM command.

Step 5.

In order to save the Master File description that will be created from the HOLD performed in Step 6, you will need to specify a location where the Master File description will be written. Since the 'ON TABLE HOLD' command in EDA normally creates a temporary file and the default behavior for EDA is to erase temporary files upon disconnection, you will need to do the following to save this file for use beyond your connected EDA session:

On MVS:

Add an allocation for ddname HOLDMAST to point to a partitioned data set. This partitioned dataset should be LRECL 80, RECFM FB). The Master File description for the HOLD data file created in Step 6 will be created as a member of this partitioned dataset. The following is an example of the DYNAM allocation:

```
DYNAM ALLOC FILE HOLDMAST DA prefix. HOLDMAST.DATA SHR REU
```

Step 6.

Create a stored procedure on the EDA Server that includes the commands from Steps 3, 4 and 5, and issues either an SQL or TABLE request to retrieve all of the columns in the original database. This request must include the date fields that DEFCENT and YRTHRESH have been applied to. You must also add code to HOLD the data on the EDA Server. The SQL request below illustrates this:

```
DYNAM ALLOC FILE DATETEMP DA prefix.DATETEMP.DATA SHR REU
DYNAM ALLOC FILE DATECOPY DA prefix.DATEFILE.DATA SHR REU
DYNAM ALLOC FILE HOLDMAST DA prefix.HOLDMAST.DATA SHR REU
SELECT I6YMDFLD, I6MDYFLD, I4YMFLD
        A6YMDFLD, A6MDYFLD, A4YMFLD
FROM DATETEMP
ORDER BY I6YMDFLD;
TABLE
ON TABLE HOLD FORMAT ALPHA AS DATETEMP
END
```

Note: The SQL request you use should include the columns, sorting and screening conditions that are required for your load procedure.

The HOLD file will be written to the destination specified in the DYNAM statement from Step 3 for ddname DATETEMP. The Master File description will be created in the destination specified in Step 5. The name used for the Master File description will be the AS name value in the 'ON TABLE HOLD' statement. In this case, the name of the Master File description will be DATETEMP. This Master File description can be used as a template for your load procedure or in creating a new Master File description for the data file after it is reloaded.

Step 7.

Use the HOLD file created in Step 6 as input to a load procedure for a new database.

Step 8.

Create a new Master File description for the new database. You can use the HOLD Master File description as a template.

Step 9.

Make the new database and Master File description available to your EDA Server environment, and test all applications against it.

Step 10.

When all testing is satisfactory, follow the steps required by your site to move the test database and new Master File description into production.

Platform Specific Considerations for EDA

VMS EDA Considerations

Year 2000 features such as DEFCENT/YRTHRESH, and subroutine awareness of these parameters, are not available in the Release 2.1.2 EDA Server for VAX/VMS or AXP/VMS. These features are, however, available in Releases 3.3.1 of the EDA Server. These features, as well as additional Year 2000 fixes and functionality, will be made available in a future release of the VMS 2.x level server.

Expiration date patches for the VMS EDA 2.1.2 servers will be made available until further notice. EDA Server Release 3.3.1 and higher do not expire.

VMS Operating System Considerations

- VAX/VMS 6.2 requires patch VAXY2K02_062
- VAX/VMS 7.1 requires patch VAXY2K01_071
- AXP/VMS 6.2 requires patch ALPY2K01_062
- AXP/VMS 7.1 requires patch ALPY2K01_071

Note: The above patches can be obtained from Compaq's Digital Products and Services Customer Support.

Testing in EDA

Many people underestimate the importance of testing in Year 2000 projects. Make testing a priority and coordinate testing with your schedule for converting your code and your data to Year 2000 capable formats. Before testing can even begin, you will need the following:

- Adequate time
- Adequate staffing
- A set of software testing tools
- A hardware and software test environment

Some of these elements will be discussed below. With adequate planning, the conversion process should go smoothly, but be sure to include time to revise the process if testing shows that any elements of the conversion process or the remediated application do not work.

Staffing

Software testing requires a skilled and experienced staff. If your organization does not have a formal QA group, formally assign testing roles and responsibilities to your staff and assign a project leader to the task. Acquire any tools and training the staff may need before the actual testing begins. The application programmers who altered the program to make it Year 2000 capable should test the applications they modified. End-users should also be an integral part of the testing. They can provide the testing team with realistic scenarios on how the application is actually used. Once testing shows that the basic conversion process works, additional testing should be done to focus on determining where it might still fail.

The EDA Server administrator can help to coordinate the testing effort by making sure that key EDA Server test environments, databases and procedures are made available to the testing group. Database administrators may need to be involved to make the necessary database resources and authorizations available as well.

Test Environments

EDA Release 4.2 offers a new feature, `-SET TESTDATE`, that is available specifically for testing future and past dates, but, ideally, your testing group will have an environment where an altered program can be tested without affecting the current production system. With MVS, a separate LPAR can be used, or a dedicated machine can be IPL'd (initial program loaded) with the desired date. MVS users can also use a SVC11 screener product such as HourGlass 2000 or TICTOC. On other platforms, such as NT and UNIX, it may be sufficient to simply designate a machine for Year 2000 testing and reset the system clock.

Whatever test environment you create should contain copies of all the applications, metadata, procedures and data to be changed, as well as those that may be affected by the change. Using current, pre-Year 2000 data, compare the original program's output to the remediated program output. The output should be identical. Then age the data by adding a set number of years to all the date fields. A utility can be used to perform the aging. Data aging is discussed further in "Testing Year 2000 Conversion," by John Patrick, in the September/October 1998 issue of *Information Builders Systems Journal*, Volume 11, Numbers 4 and 5, page 71. The various system dates should be tested with the aged data.

The testing process should include batch jobs, as well as online testing. Some testing can be done directly on the EDA Server in a batch-like fashion, where appropriate, using `EDADEBUG` (check your EDA Server manual for details), but testing should also be performed from all front-end applications affected by any changes made to either the application or EDA Server. Test your most important applications first, then your secondary ones.

Test Dates

Contact each application's primary users to determine the best sampling of data to test on the revised application. A copy of the production datasets, along with aged data, should also be used to ensure that the modified code still works.

When testing with different centuries and calculations, keep in mind that 1900 was not a leap year, but 2000 is. Only centuries evenly divisible by 400, such as 1600 and 2000, are leap years. Many programs designed with leap year algorithms will not recognize 2000 as a leap year. You will need to manually verify the results.

There are no industry standards for required test dates. Optimally, you would test every single day for each year prior to 2000, the year 2000, and after 2000. However, time and resources are limited. There are different dates to take into consideration for applications, hardware and systems software. The following section lists specific dates that should not be excluded.

Test Dates for Applications

December 31, 1998, Thursday

January 1, 1999, Friday

January 4, 1999, Monday

These dates test whether Year 2000 changes that took place in 1998 will still work in 1999. January 4, is the first business day in 1999, as January 1 is a Friday.

September 9, 1999, Thursday

In many systems, 99/99/99 and 09/09/99 are the default values for missing dates. Therefore, make sure that these values only appear where they are intended.

December 31, 1999, Friday

This is the last day of the millennium. Processing this date, as well as the processing of data on this date, will set up some of the data to be handled on the first day of 2000.

January 1, 2000 Saturday

This is the first day of the new millennium. Although it is not a business processing day for most applications, the processing of this date, as well as the processing of data on this date, will also set up some of the data to be handled on the first business day of 2000.

January 2, 2000, Sunday

As with January 1, January 2 is not a business day, but computers will process this date to prepare for Monday, January 3, the first business day of the new millennium.

January 3, 2000, Monday

This is either the first business day of 2000 or a bank holiday, depending on the country.

January 4, 2000, Tuesday

In some countries, this will be the first business day of 2000. In others, it will be the first online day following the first batch run of 2000.

January 5, 2000, Wednesday

If January 4th was the first business day for any part of the application processing, this will be the first online day following the first batch run of 2000.

January 31, 2000, Monday

The first month's end in 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

February 29, 2000 is a leap day. Many systems fail every leap year, on the leap day, due to a lack of testing on the leap year determination code. A year is a leap year if divisible by 4, unless it is a century year, in which case it must be divisible by 400. For example, 1700, 1800, and 1900 are not leap years, but 2000 is.

March 31, 2000, Friday

The end of the first quarter of 2000.

January 2, 2001, Tuesday

The first business day of 2001. Testing data for this day will determine whether the first year-end of the new millennium works.

Test Dates for Hardware:

These dates were chosen to verify that hardware components will retain the correct time and date after crossing over the Year 2000 boundary. Hardware should maintain current crossovers, such as 1998 to 1999, as well.

December 31, 1998, Thursday

January 1, 1999, Friday

Testing the date crossover from year-end 1998 to the start of 1999.

December 31, 1999, Friday

January 1, 2000, Saturday

This is the crossover from year-end 1999 to the first day of 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

This is the crossover for next leap day.

Test Dates for Systems Software:

Systems software includes operating systems, database management systems and so on. These dates were chosen to verify that the system software will retain the correct time and date after crossing the Year 2000 boundary. It should also maintain current crossovers as well.

December 31, 1998, Thursday

January 1, 1999, Friday

Testing the date crossover from year-end 1998 to the start of 1999.

December 31, 1999, Friday

January 1, 2000, Saturday

This is the crossover from year-end 1999 to the first day of 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

These dates are necessary for testing the crossover for the leap day on February 29, 2000.

Summary of Testing Issues

Testing Checklist

- Identify each application's resources.
- Identify target files for update, browse, etc.
- Generate initial test data to reflect a subset of production environment.
- Establish documentation procedures.
- Allow time for review of first application's results.
- Repeat process and roll out.

Source Control and Management

- Track code prior to conversion.
- Establish a baseline and audit trail procedure.
- Formalize an approval and movement process.
- Automate deployment.
- Build in a contingency plan.
- Satisfy internal auditing.
- Develop a production code change procedure while the code is in remediation.

Unit Testing Tasks

- Program and transaction level verification.
- Identify transaction flows for online applications.
- Black box testing, also known as functional testing.
- Keep test files and database sizes manageable.
- Create valid, invalid and unexpected test cases.
- Define what the acceptable criteria is for testing.
- Use the criteria as a building block for expanded testing.

Business Cycle Testing

- Date simulation during real-time updates.
- End-of-month, quarter and fiscal year cycle.
- Create data to trap timelines and totals.
- Consider running the current and test systems in parallel for one cycle.
- The test data set used here may be larger than the data set used in other test cases.
- Encourage your organization's business analysts to participate.

Test Coverage

- Determine how much code has been "exercised" during a specific test.
- Track untested lines of code.
- Calculate the percentage of code that has been tested.
- Make sure code that has been changed is tested.

Other Considerations

- Assess the way that date data imported from external systems is being described to EDA in the Master File description. Is the data stored with four- or two-digit years?
- Is your data being exported to other systems? Do exported dates contain the right century? These questions may also apply to HOLD files. If you perform an ON TABLE HOLD FORMAT ALPHA with the DEFCENT/YRTHRESH parameters set, the century portion does not get saved when holding Smart Date field formats of MDY, DMY, and YMD.

Summary of New Date Features

Release 3.3.1

SET DEFCENT=19

SET YRTHRESH=00

Set at the global level.

SET DEFCENT and SET YRTHRESH available at the file and field level

FDFCENT/FDFC and FYRTHRESH/FYRT (file level)

DEFCENT/DFC and YRTHRESH/YRT (field level)

Subroutines will use DEFCENT and YRTHRESH, and the last argument has been expanded.

Date Handling for the Year 2000 In EDA

The default date format display in EDA is now MMDDCCYY. This applies to all areas within EDA that displays a date:

? FILE

? FDT

MODIFY FILE FN

CREATE FILE FN

SET DATEFNS=ON/OFF

EDA functions: YMD, DMY, and MDY are now DEFCENT/YRTHRESH aware.

Release 4.2.1

REBUILD / DATE NEW

SET DATEDISPLAY = ON/OFF

Controls display of base date: 12/31/00

CHECK FILE fn HOLD ALL

Writes DFC/YRT to Hold Master

ON TABLE HOLD

Writes DFC/YRT to Hold Master

New Date Math Functions for Year 2000:

Summary of New Date Features

DATEADD (Smart Date, 'unit', #units)
SET BUSDAYS = {day-list/_MTWTF_
SET HDAY=STKM

DATEDIF(FR_smartdate,TO_smart,'unit');
DATEMOV(smartdate, 'move-point',outdt);
DATECVT(date, 'oldate', 'newformat')

Displaying Invalid Smart Dates in Reports:

SET ALLOWCVTERR={ON|OFF}

OFF: invalid date gives format conversion error

ON: MISSING=ON – invalid date is MISSING

ON: MISSING=OFF – invalid date is base date

YRTHRESH = -n

YRTHRESH can now be set as an offset from the current year as well as specifying a year.

TODAY Subroutine enhancement

-SET &HELLO=TODAY('A10');

Display &Date variable without separators

&YYMD

&MDYY

&DMYY

YYJUL – Julian Format

SET TESTDATE={yyyymmdd/TODAY}

This allows you to set the test date within EDA for testing purposes only.

EDA log changes

EDAPRINT log now displays with four-digit years.

Frequently Asked Questions

1. What releases of the EDA/API, EDA/Client or EDA/ODBC Extender software are Year 2000 capable?

Year 2000 capability is not an issue for either the EDA/API, EDA/Client or EDA/ODBC Extender, as these products do not directly manipulate data. The EDA/Client software is used to connect to, and transfer data to and from, an EDA Server. The EDA/API and EDA/ODBC Extender provide function calls that facilitate communication between the application and the EDA Server. Year 2000 capability is only relevant for the client application and EDA Server, where data storage, display and/or manipulation can occur.

2. Will expiration date zaps be provided to extend the expiration of older EDA Server releases like the UNIX EDA 1.x and MVS EDA 2.x servers in the Year 2000?

EDA Server releases stopped expiring with release 3.1. EDA Server Release 3.1 and higher do not expire. Existing expiration date zaps created before the Year 2000 will be provided, if needed, for earlier releases. However, no new expiration date zaps will be created in the Year 2000 for these servers. Information Builders recommends that users running releases prior to 3.1 upgrade their environments.

3. What release of the EDA Server is considered to be the Year 2000 capable release?

On platforms like MVS, UNIX, NT and VM, Release 3.3.1 of the EDA Server is considered to be the Year 2000 capable release. To obtain more information on the current version of Information Builders Year 2000 Compliance criteria for these and other products, visit the Information Builders web site at:

http://www.ibi.com/solutions/y2k/y2k_intro_index.html

This document is updated regularly and is available for download from this web site along with other Year 2000 information.

4. What is the first release of the EDA Server that supports the DEFCENT and YRTHRESH parameters?

Release 3.3.1 is the first release of the EDA Server that supports the DEFCENT and YRTHRESH parameters. These features are documented in the server manuals for EDA Release 3.3.1 and higher.

5. To what field formats can I apply DEFCEM and YRTHRESH?

DEFCEM and YRTHRESH can be applied to date fields that do not provide the century portion, and where date conversion takes place. Date conversion takes place when:

- Converting old dates to Smart Dates and the first two digits are not supplied.
- Converting from ACTUAL to USAGE and the first two digits are not supplied.
- Data entry is performed, via FIXFORM, and the first two digits are not provided.

6. What happens if you code DEFCEM and YRTHRESH, but the field already contains the century?

If the century is explicit, DEFCEM and YRTHRESH will be ignored.

7. Is there an expiration date for an MVS Release 3.3.1 server?

Expiration of the EDA Server was eliminated beginning in Release 3.1. All EDA Server 3.1 and higher releases do not expire.

8. How can I get the current system date returned, including the century, when &DATE only returns a two-digit year along with the month and day?

In EDA Server Release 3.1 and higher, the Dialogue Manager variable &DATEMDYY will return the system date in the same order as &DATE but will also include the century. Other variables available:

&DATEYYMD

is today's date shown as yyyy/mm/dd

&DATEMDYY

is today's date shown as mm/dd/yyyy

&DATEDMY

is today's date shown as dd/mm/yyyy

&DATEMtrDYY

returns the translated month value (i.e. December 15, 1998)

9. What is the base date EDA uses for Smart Date format fields?

If the Smart Date format contains year, month and day components, then the base date is December 31, 1900. If the format uses only a month and a year, then the base date is January, 1901.

EDA

5 Enterprise Copy Server

Year 2000 Capable Releases and Enhancements

Enterprise Copy Server Release 3.3.2a is considered Year 2000 capable. The metadata, or Master File descriptions, for all Copy Manager internal tables have been updated to handle four-digit years. In addition, all procedures that reference date columns have been updated to use four-digit years.

Copy Client Release 3.3 is considered Year 2000 capable. Release 3.3 and higher of Copy Client are supported for use with the Year 2000 capable release of Copy Server, Release 3.3.2a. Copy Client will now check if the Copy Server in use is the Year 2000 capable version. If so, it will send four-digit years for all date values.

Non-Year 2000 Capable Releases

If a copy request was created in a non-Year 2000 capable release of the Copy Server or Copy Client, the following steps will make these copy requests Year 2000 compliant:

- Open the copy requests in the Year 2000 capable release of the Copy Client. Since the older versions of Copy Client did not have the ability to schedule requests past December 31, 1999, you'll now be able to reschedule the requests as desired.
- Save the requests.

If a user is running a non-Year 2000 capable release of Copy Server or Copy Client on January 1, 2000, all aspects of the Copy Manager environment will be impacted. Pre-Year 2000 capable releases of the Copy Client cannot schedule copy requests beyond December 31, 1999, and the Copy Manager internal tables in the pre-Year 2000 release of the Copy Server do not support four-digit years. Therefore, upgrading to the Year-2000 capable release of the Copy Client and Server is mandatory for support of the Copy Manager environment beyond 1999. The following specific areas will be affected:

- Scheduled copy requests will not run because copy requests could not be scheduled into 2000 on the pre-Year 2000 capable Copy Client release.
- Copy Client will not be able to connect to the Copy Server, because the dates in the internal tables will be incorrect.

Year 2000 Testing for Copy Manager Applications

Set the system date to a test date on both the client and server systems. For a new Copy Server, run CMIVP in the Year 2000 capable Copy Server release. Schedule copy requests and ensure that they run. For an environment with existing Copy requests, run CMDRPLOG in the Year 2000 Copy Server release. Schedule copy requests and ensure that they run.

Note: CMIVP will drop and recreate the Copy Manager internal tables and should **not** be used in an environment with existing copy requests that must be saved. CMDRPLOG should be run for environments migrating from a pre-Year 2000 capable release of the Copy Server to ensure the availability of the Year 2000 capable Copy Manager internal tables.

Following are some suggested dates for Year 2000 testing of Copy Manager applications:

December 31, 1998, Thursday

January 1, 1999, Friday

January 4, 1999, Monday

These dates test whether Year 2000 changes that took place in 1998 will still work in 1999. January 4 is the first business day in 1999, as January 1 is a Friday.

September 9, 1999, Thursday

In many systems, 99/99/99 and 09/09/99 are the default values for missing dates. Therefore, make sure that these values only appear where they are intended.

December 31, 1999, Friday

This is the last day of 1999. Processing this date, as well as the processing of data on this date, will set up some of the data to be handled on the first day of 2000.

January 1, 2000, Saturday

This is the first day of the new millennium. Although it is not a business processing day for most applications, the processing of this date, as well as the processing of data on this date, will also set up some of the data to be handled on the first business day of 2000.

January 2, 2000, Sunday

As with January 1, January 2 is not a business day, but computers will process this date to prepare for Monday, January 3, the first business day of the new millennium.

January 3, 2000, Monday

This is either the first business day of 2000 or a bank holiday, depending on the country.

January 4, 2000, Tuesday

In some countries, this will be the first business day of 2000. In others, it will be the first online day following the first batch run of 2000.

January 5, 2000, Wednesday

Year 2000 Testing for Copy Manager Applications

If January 4th was the first business day for any part of the application processing, this will be the first online day following the first batch run of 2000.

January 31, 2000, Monday

The first month's end in 2000.

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

February 29, 2000 is a leap day. Many systems fail every leap year, on the leap day, due to a lack of testing on the leap year determination code. A year is a leap year if divisible by 4, unless it is a century year, in which case it must be divisible by 400. For example, 1700, 1800, and 1900 are not leap years, but 2000 is.

March 31, 2000, Friday

The end of the first quarter of 2000.

January 2, 2001, Tuesday

The first business day of 2001. Testing data for this day will determine whether the first year-end of the new millennium works.

Enterprise Copy Server

6 FOCUS for UNIX

Year 2000 Capable Releases

Commands

The most recent release of FOCUS for UNIX, Release 6.8.5, is Year 2000 capable. Release 6.8.5 of FOCUS for UNIX is functionally equivalent to Release 7.0.8 of FOCUS for IBM Mainframe and includes the DEFCENT and YRTHRESH commands. DEFCENT sets a default century for dates with two-digit year fields. YRTHRESH sets the century value based on a 100-year span of time set by the user.

Release 6.8.5 is the first release of FOCUS for UNIX to incorporate DEFCENT and YRTHRESH. It satisfies the criteria for all the major categories of data integrity: General Integrity, Date Integrity, Explicit Century, and Implicit Century across most UNIX platforms. For an explanation of these categories, see the glossary or Information Builders Year 2000 Compliance Issues Statement (www.ibi.com/solutions/y2k/y2k_index.html).

Those UNIX platforms that Release 6.8.5 doesn't support will be supported by Release 6.8.6, which will be the functional equivalent of Release 7.0.8R of FOCUS for IBM Mainframe.

Frequently Asked Questions

The following questions and answers concern Release 6.8.5 of FOCUS for UNIX.

1. Are you aware of any Year 2000 specific bugs affecting Release 6.8.5 of FOCUS for UNIX?

The Open Systems Division is not currently aware of any Year 2000 specific bugs or problems affecting Release 6.8.5 of FOCUS for UNIX.

2. Are you aware of any &variable issues affecting Release 6.8.5 of FOCUS for UNIX?

The Open Systems Division is not currently aware of any &variable issues with Releases 6.8.5. &YMD and &DATE will work within both of these releases after December 31, 1999. Release 6.8.6, the next release of FOCUS for UNIX, will feature &YYMD and similar variables that feature four digit year values.

3. Are there any new Year 2000 related subroutines?

Release 6.8.5 includes the TODAYYY subroutine, which returns the current day with a four-digit year.

4. Are you aware of any platform-specific issues affecting Release 6.8.5 of FOCUS for UNIX?

The Open Systems Division is not currently aware of any platform- or product-specific issues regarding FOCUS for UNIX and Year 2000.

5. Where can I find more Year 2000 related documentation?

For further Year 2000 related documentation, check the FOCUS for UNIX Release Notes at http://techsupport.ibi.com/support/unx/unx_rln_toc.html, especially the following: “Release Notes for FOCUS for Open Environments,” “Date Handling for the Year 2000,” and “Year 2000 Date Subroutines.”

Non-Year 2000 Capable Releases

Commands

Users of earlier, non-Year 2000 capable releases of FOCUS for UNIX can apply ZAPEXP to eliminate the expiration date from their copy of FOCUS. They can also apply the General Data Integrity patch, 6.5.5. This patch incorporates the CLOCKIT fix, which enables older releases of FOCUS for UNIX to handle date values for 2000 and beyond.

Users who do not apply ZAPEXP will see their releases expire on January 1, 1999. Users who apply the zap, but do not apply the 6.5.5 patch, will receive incorrect year values for dates in the next century. Based on the manner in which older versions of FOCUS for UNIX handle dates, the years 2000 to 2009 will return a value of 10, the years 2010 to 2019 will return a value of 11, and so on.

Frequently Asked Questions

The following questions and answers are regarding users of FOCUS for UNIX who have applied ZAPEXP and the 6.5.5 patch. In each case, the question assumes that the user is attempting to operate their system after January 1, 2000.

1. Will users be able to pass Year 2000 dates to the DBMS?

Users will only be able to pass dates to a FOCUS DBMS, not a third-party DBMS.

2. Will the data access engine still function?

Yes, the data access engine will still function.

3. Will subroutines that read the system clock, such as TODAY and HHMMSS, display the correct, current, Year 2000 date?

The following releases of FOCUS for UNIX will expire on January 1, 1999:

FOCUS for UNIX Release 5.5

FOCUS for UNIX Release 6.5, 6.5.1, 6.5.1m

FOCUS for UNIX Release 6.8.

Users who have only applied ZAPEXP to these releases will not see the correct system date past December 31, 1999. Users who have upgraded to Release 6.5.5 will see the correct date in the new century.

4. Are there any &variable issues with Release 6.5.5?

No. Year 2000 improvements have been made to &variables in Release 6.5.5. Amper variables, such as &YMD, will continue to use two-digit years and will be susceptible to Year 2000 problems.

5. Are there any new Year 2000 subroutines?

FOCUS for UNIX Release 6.5.5 include DA and DT subroutines. DA subroutines convert dates to the corresponding number of days that have elapsed since January 1, 1900. DT subroutines converts these numeric values back to their original date formats.

6. Will users be able to write trace or log files if tracing is necessary?

Yes, users will be able to write trace or log files.

7. Are you aware of any issues with 3rd party communications that might be affected?

No. The Open Systems Division is not currently aware of any issues with 3rd party communications that might be affected by running a release of FOCUS for UNIX prior to Release 6.8.5.

8. Are you aware of any issues with front-end application tools that might be affected?

No. The Open Systems Division is not currently aware of any issues with front-end application tools that might be affected by running a release of FOCUS for UNIX prior to Release 6.8.5.

9. Are you aware of any issues where 3rd party hardware or operating system software might be affected by FOCUS for UNIX?

No, the Open Systems Division is not currently aware of any issues where 3rd hardware or operating system software might be affected by running a release of FOCUS for UNIX prior to Release 6.8.5.

Testing and Other Issues

The Open Systems Division does not currently recommend a particular product or method for Year 2000 testing of FOCUS for UNIX. However, a simple method for testing a system's Year 2000 compatibility is to use the SET SYSTEMDATE function to reset the system clock for a date beyond December 31, 1999 and test sample data.

Leap years are a significant consideration for Year 2000 capability testing. The Open Systems Division recommends that users test their systems with the following dates:

- 02/29/2000
- 02/29/1996
- 02/29/1900

The first two are valid dates, but the third is not. Since 1900 was not a leap year, it should not be accepted by the application. Also, see the section of this publication devoted to testing on FOCUS for IBM Mainframe.

Third Party Issues

The Open Systems Division recommends that all users contact the vendors of 3rd party front-end, communications, hardware, and software products for information on the patches and upgrades they have available.

FOCUS for UNIX has no special features for detecting or correcting non-compliant dates provided by a 3rd party vendor.

System Date

FOCUS for UNIX reads the system date from the CLOCKIT function.

Environmental Issues

The Open Systems Division is not currently aware of any other environmental issues, such as system logicals, executables and so forth, that might affect the Year 2000 capabilities of FOCUS for Unix.

For Further Information

The Open Systems Division is not currently aware of any other Year 2000 issues or concerns that might affect your product that were not explicitly described above.

For further information on Year 2000 data integrity issues, see Information Builders Year 2000 Compliance Issues Statement: http://www.ibi.com/solutions/y2k/y2k_intro_index.html

FOCUS for UNIX

7 WebFOCUS Server

WebFOCUS Year 2000 Capable Releases and Enhancements

The WebFOCUS server has the same numerical release level as the EDA level with which it operates. For example, WebFOCUS 3.3.1 has the same capability to deal with Year 2000 issues as EDA 3.3.1. This means that features such as the sliding date window are available in the WebFOCUS Server Release 3.3.1 and higher. These features can enable your program to infer the correct century for date elements where century value is absent or ambiguous. See the section, *How to Remediate* (within Chapter 4, *Remediation in EDA*), for a more detailed explanation of the sliding date window.

WebFOCUS Server

8 FOCUS on OpenVMS

Year 2000 Compatible Software and Hardware

IBI Year 2000 Compatible Software Available for VMS

OS / Platform	First Year 2000 Compatible Release	Subsequent Year 2000 Compatible Releases
OpenVMS Alpha	6.9	6.9.1, 6.9.2, 6.9.3, 6.9.4
OpenVMS VAX	6.9	6.9.1, 6.9.2, 6.9.3, 6.9.4

Year 2000 Compatible Hardware Platforms Available for VMS

Platform/OS	Version	OS Patches for Y2K
VAX / OpenVMS	5.5-2	
	6.2	VAXY2K02_062
	7.1	VAXY2K01_071
Alpha / OpenVMS	6.2	ALPY2K01_062
	7.1	ALPY2K01_071

Year 2000 Commands and Features for OpenVMS

Command	First available in release
New Y2K compatible FUSELIB subroutines	6.9.3
SET DATEFNS=ON/OFF	6.9.3
SET TESTDATE=date	6.9.4
SET FYRTHRESH, FDEFCENT (file level)	6.9
SET DEFCENT, YRTHRESH (application or field level)	6.9

Year 2000 Remediation

Year 2000 Bugs

Currently, there are no open Year 2000 related bugs for FOCUS on VAX or Alpha.

Year 2000 Subroutine Issues

In Release 6.9.3, the following subroutines: YMD, DMY, and MDY, return -99999 for invalid date input. This differed from previous behavior in FOCUS for OpenVMS and IBM Mainframe behavior. The subroutines were changed, and in Release 6.9.4, these subroutines once again return 0 for invalid date input.

Sample Code Techniques for Year 2000 Remediation

This example illustrates the use of the third parameter in the subroutine call. The input field is in an I6YMD format that contains two-digit years. Using 1950 as a pivot year for the default century, this program converts the data to a full four-digit year form in an 'I8' format.

```
SET DATEFNS = ON
SET DEFCENT=19,YRTHRESH=50
TABLE FILE DATE
ON TABLE SUBHEAD
" THIS EXAMPLE ILLUSTRATES THE USE OF THE THIRD PARAMETER IN THE      "
" SUBROUTINE CALL.  THE INPUT FIELD IS AN I6YMD FORMAT.  IT CONTAINS  "
" 2 DIGIT YEARS.  SETTING A PIVOT YEAR OF 1950 RESULTS IN CONVERSION  "
" TO A FULL 4 DIGIT YEAR VIA THE FORMAT OF 'I8'.                      "
PRINT D2_I6YMD AND COMPUTE
X/I8YYMD=AYMD(D2_I6YMD,1,'I8');
END
```

Output:

```
PAGE      1

THIS EXAMPLE ILLUSTRATES THE USE OF THE THIRD PARAMETER IN THE
SUBROUTINE CALL.  THE INPUT FIELD IS AN I6YMD FORMAT.  IT CONTAINS
2 DIGIT YEARS.  SETTING A PIVOT YEAR OF 1950 RESULTS IN CONVERSION
TO A FULL 4 DIGIT YEAR VIA THE FORMAT OF 'I8'.
D2_I6YMD      X
-----      -
97/09/16      1997/09/17
00/02/29      2000/03/01
01/02/28      2001/03/01
00/02/28      2000/02/29
```

Environmental Issues and Year 2000 Compliance

There are no environmental issues, such as system logicals, executables, and so forth, that are known to affect Y2K compliance for FOCUS for OpenVMS.

Year 2000 Testing

Year 2000 Testing by IBI

Currently, the OpenVMS division is not using any products or methods to reset the system date for Year 2000 testing.

Year 2000 Testing by Customers

If you are using Release 6.9.3 or higher, use the following functions:

- SET TESTDATE (see usage below).
- SET YRTHRESH, SET DEFCENT (see usage in “Sample Code Techniques for Year 2000 Remediation” above).

TESTDATE is a new SET command that allows you to temporarily alter your FOCUS system date for a given application program to any date. This allows you to determine what impact the Year 2000 will have on your application programs.

Note: Only use TESTDATE for testing purposes with a test database.

The syntax for TESTDATE is:

```
SET TESTDATE = YYYYMMDD
```

where:

```
YYYYMMDD
```

Is an 8-digit date in the format YYYYMMDD.

The value of TESTDATE affects all reserved variables that retrieve the current date from the system. Setting TESTDATE also affects any place in FOCUS where a date is used (CREATE, MODIFY, etc.), but does not affect the date referenced directly from the system. TESTDATE can be equal to YYYYMMDD.

Alternately, if the system can be used exclusively for testing, users can SET TIME, like this:

```
$ SET TIME=19-JAN-2000:19:31:0.0
```

Leap Year Testing by Customers

Information Builders recommends that customers use the following dates to test their systems for agreement and compliance with leap years in the current and next centuries:

February 28, 2000, Monday

February 29, 2000, Tuesday

March 1, 2000, Wednesday

February 28, 2000 is the first leap day of the new millennium. Many systems fail every leap year on leap day due to lack of testing on the leap year determination code. A year is a leap year if divisible by 4, unless it is a century year, in which case, it must be divisible by 400. For example 1700, 1800, and 1900 are not leap years, but 2000 is.

Additional Year 2000 Test Dates

Information Builders recommends that customers use the following additional dates to test their systems' ability to deal with dates in the next century:

December 31, 1998, Thursday

January 1, 1999, Friday

January 4, 1999, Monday

Testing with these dates ensures that Year 2000 changes that took place in 1998 will still work in 1999. January 4 is the first business day in 1999, as January 1 is a Friday.

September 9, 1999, Thursday

In many systems, 99/99/99 and 09/09/99 are default values for missing dates. Make sure that these values occur only where intended.

December 31, 1999, Friday

This is the last day of the millennium.

January 1, 2000, Saturday

Although this is the first day of 2000, it is not a business processing day for most applications. However, the data may be used in processing performed on the first business day of 2000.

January 2, 2000, Sunday

As with January 1, January 2 is not a business day, but computers will process this date to prepare for Monday, January 3, the first business day of the new millennium.

January 3, 2000, Monday

Varies by country. It is either the first business day of 2000, or a bank holiday.

January 4, 2000, Tuesday

In some countries, this will be the first business day of 2000 or the first online day following the first batch run of 2000.

January 5, 2000, Wednesday

If January 4th was the first business day for any part of the application processing, this will be the first online day following the first batch run of 2000.

January 31, 2000, Monday

The first month's end in 2000.

March 31, 2000, Friday

The first end-of-quarter in 2000.

January 2, 2001, Tuesday

The first business day of 2001, to ensure that the first year-end of 2000 works.

Non-Year-2000 Capable Releases

Release 6.8.6

This release can be used without any problems, if:

- User's applications are **not** using FUSELIB Date subroutines.
- User's applications are **not** using FOCUS database with 6 digit (I6YMD, A6YMD) old dates or 4 digit (I4YM, A4YM) old dates.
- For example, user is running FOCUS databases with Smart Dates only (no old dates).
- For example, user is running non-FOCUS databases (Oracle, Sybase, Ingres, etc.).

On and after January 1, 2000:

- Users will be able to pass Year 2000 dates to the DBMS if the DBMS is Year 2000 compatible.
- The data access engine will function.
- Attach management services will work for EDA only.
- Subroutines that read the system clock, such as TODAY and HHMMSS, will display the correct date.
- Users will still be able to write trace or log files with tracing.
- Users must use the correct, Year 2000 compatible version of all 3rd party database products (for example, RDB, PROGRESS, etc.).
- The FOCUS front-application tools: MODIFYTALK, TABLETALK, AUTO-utilities, and FILETALK are all Year 2000 compatible.

Currently, Information Builders is unaware of any issues with 3rd party hardware or operating system software that might be affected (for example, RMS, DECNET, UCX).

Release 6.8.5 and below

These releases **will not function properly** after Dec. 31, 1999 because the CLOCKIT routine has a bug in the way dates are displayed for system variables (for example, &MDY), auto modules (for example, MODIFYTALK), and query commands (for example, ? FDT).

Will any patches be provided to extend the expiration dates beyond the Year 2000?

- FOCUS for OpenVMS releases 6.8.6 and above for VAX and Alpha have been patched so they will never expire.
- FOCUS for Alpha/OpenVMS releases 6.8.5 and below will expire. Users can request a patch to eliminate the expiration, but the product will not function properly due to the CLOCKIT bug described above.
- FOCUS for VAX/OpenVMS releases 6.8.5 and below have already been patched so that they will not expire, but the product will not function properly due to the CLOCKIT bug described above.

Third Party Issues

Third Party Databases

Each database product that we support has a Year 2000 compatible release. In most cases, we have certified FOCUS with this release of the database product. (There are some remaining cases in which FOCUS has not yet been certified, but we are planning to update those, as well.) The following table of database products lists their Year 2000 compatible versions and whether they have FOCUS certification.

VAX

Database product	Year 2000 compatible versions*	Certified with FOCUS?
Software AG's Adabas A	No information available	No
Oracle's CODASYL DBMS	4.2, 6.0	Yes
Intersystem's DSM (Digital Standard Mumps)	7.1	Not yet

Third Party Issues

CA's Ingres	6.4/05 with patch #5126 for Alpha	Not yet
CA's OpenIngres	1.2 (all versions are compatible)	Yes
Oracle's Oracle RDMS	7.1	Yes
Progress	6.2N03 and 8.1A03	Yes
Oracle's Rdb RDMS	4.0*, 4.1*, 4.2*, 5.1, 6.0, 6.1, 7.0	Yes
Sybase SQL Server	10.0.3	No
Teradata	No information available	No

* The versions listed are those that we have in-house. We have not installed or tested this version yet. Later versions of these products will also be Year 2000 compatible.

Alpha

Database product	Year 2000 compatible versions*	Certified with FOCUS?
Software AG's Adabas A	No information available	No
Oracle's Codasyl DBMS	6.1, 7.0	Yes
Intersystem's DSM (Digital Standard Mumps)	7.1	Not yet
CA's Ingres	6.4/05 with patch #5126 for Alpha	Not yet
CA's OpenIngres	1.2 (all versions are compatible)	Yes

FOCUS on OpenVMS

Oracle's Oracle RDMS	7.1, 7.2, 7.3, 8.0	Yes
Progress	6.2N05 and 8.1A03	Yes
Oracle's Rdb RDMS	5.1, 6.0, 6.1, 7.0	Yes
Sybase SQL Server	10.0.3	No

* The versions listed are those that we have in-house. We may not have installed or tested these yet. Later versions of these products will also be Year 2000 Compatible.

Third-Party Compatibility Issues

Currently, IBI is not aware of any compatibility issues with any 3rd party products, including front-end, communications, hardware, and software, that might affect Year 2000 compliance with FOCUS for OpenVMS on VAX and Alpha in the Release 6.9 or higher.

For Further Year 2000 Information

To access further Year 2000 information regarding FOCUS for OpenVMS on VAX and Alpha, contact the website: http://ibisparc3:9010/OSD_Web/P2000.html.

9 FOCUS Desktop

Determining If You Have a Year 2000 Problem

Neither FOCUS Six nor FOCUS Desktop provide an automatic method to determine where Year 2000 exposure exists within an application. An article entitled “Y2KCHECK: A FOCUS Master File description Scanning Utility,” by Chris Bell in the September/October 1998 issue of *Information Builders Systems Journal*, Volume 11, Numbers 4 and 5, page 165, explains how FOCUS can be used to scan your application for date fields that are not in Smart Date format. The utility Y2KCHECK was written for use with Mainframe FOCUS running on either MVS or CMS, but can easily be adjusted for use on the PC.

Remediation

Remediation In FOCUS For Windows (FFW)

FOCUS for Windows offers both a 16-bit and a 32-bit solution to your Year 2000 compliance issues. The 16-bit solution (FOCUS Six, Release 6.4x) provides “explicit century” support, while FOCUS Desktop (32-bit solution) provides both “explicit “ and “implicit” century support. FOCUS Desktop is a relatively new product, and most users are currently using FOCUS Six.

Remediation In FFW 6.4x

FOCUS Six Release 6.40 supports general data integrity when running on any supported version of Windows; FOCUS Six release 6.40 will work correctly regardless of the date setting on the PC’s internal clock, until Dec. 31, 2038. In this respect, FOCUS Six Release 6.40 is like many C-based applications. Because of a problem with the internal storage of dates, applications written in C and compiled using the Microsoft 16-bit C compiler will cause many 16-bit Windows applications (including FOCUS Six) to fail after Dec. 31, 2038. There is no known work in existence for this issue. Applications written using the 32-bit compiler use a different method for storing dates and will not be affected by this limitation.

Developers who are migrating their applications to make them Year 2000 compliant must make changes to *both* their applications *and* their databases to make them Year 2000 compliant on FOCUS Six Release 6.40. Furthermore:

- If databases contain partial date containers in storage, the database fields must be modified using logic set up by the developer to bring them into compliance.
- If applications represent, at any point in the chain of processing, a date in less than a full date container, the application logic must be changed to use a full date container.

FOCUS Desktop

Information Builders cannot guarantee that FOCUS Six 6.4x will correctly infer missing century information from applications that were created prior to Release 6.40. If a date is represented as 01/01/99, there is no guarantee that *any* release of FOCUS Six will be able to infer the correct century, because it will assume that the century value is 19. To avoid this issue, always use a full date container by specifying either:

- A day, month and full 4-digit year (for example, a Gregorian-style date).
- A day-of-year and full 4-digit year (for example, a Julian-style date).

FOCUS Six release 6.40 provides two additional year format designators that work with DA and DT prefixed functions, and it supports full date containers as follows:

YY

Examples: DTMDYY, DTYYMD, DAYYDM, etc.

As a convenience for developers performing Year 2000 conversions, you can now explicitly state in the function call that the function takes a four-digit year. It isn't required that you rename function calls; the new date format-naming function "aliases" the new date format designator YY designator to the standard function. However, YY coding can be used to indicate that functions have been recast, and are now passing full date containers.

Note: Using these functions with a Y or YY format in the function name and passing a date containing a two-digit year will assume a 20th century date.

For example:

- If you pass 010199 to a FUSELIB date function, FOCUS assumes you mean the date to be 01/01/1999.
- If you pass 010100 to the function, FOCUS assumes you mean the date to be 01/01/1900.

CY

Examples: DTMDCY, DTCYMD, DACYDM, etc.

CY is a new date format designator that lets you specify a century for a two-digit year according to rules you input: SET YRTHRS= and SET DFCENT=. It's specifically designed to ease date conversions on legacy databases.

If your database contains historical dates, but represents the dates with partial date containers (for example, 1898 represented as '98'), you could convert your data as follows:

1. Create a new database that uses full date containers.
2. Write a conversion MODIFY FOCEXEC that reads six-digit dates from your existing databases, and SET the DFCENT to 18 and the YRTHRS to 50.
3. Include a call to use the DAxxx and DTxxx with the CY date format designator to convert the date to a full date container.
4. Output the dates to your new database.

The CY date designator and the companion SET variables (SET DFCENT and SET YRTHRS) provide a sliding window, but only work with the DA and DT functions. Note that they do not provide general implicit date support in FOCUS Six and are not to be confused with the FOCUS Desktop implicit date support settings (SET DEFCENT and SET YRTHRESH).

Note: There is a difference in the base dates between FUSELIB functions and FOCUS Smart Dates. There is a one-year (365-day) difference in the "base date" between core FOCUS and the FUSELIB date functions as follows:

- Base date for FOCUS Six Smart Dates is 01/01/1901.
- Base date for FUSELIB is 01/01/1900.

For example:

- If you are passing a Smart Date to a DTxxx function, you need to add 365 to it before passing it.
- If you are creating a Smart Date from a DTxxx function, you must subtract 365 from the value DT outputs.

Amper Variables

New date amper variables, also known as &variables, have been added to FOCUS Six, Release 6.4x. These are based on the existing date amper variables, but now return a full date container. They are as follows:

New Amper Variable	Description
&MDYY	Date that you started the FOCUS session. Displayed as MMDDYYYY.

FOCUS Desktop

&DMYY	Date that you started the FOCUS session. Displayed as DDMMYYYY.
&YYMD	Date that you started the FOCUS session. Displayed as YYYYMMDD.
&FOCSYSMDYY	Current (system clock) date. Displayed as MMDDYYYY.
&FOCSYSDMYY	Current (system clock) date. Displayed as DDMMYYYY.
&FOCSYSYYMD	Current (system clock) date. Displayed as YYYYMMDD.

Converting Your Application's Date Variables

Old Amper Variable	Description	For Y2K capability, use
&DATE	Date that you started the FOCUS session. Displayed as a string: MM/DD/YY.	&MDYY
&FOCSYSDATE	Current (system clock) date. Displayed as MM/DD/YY.	&FOCSYSMDYY
&FOCSYSMDY	Current (system clock) date. Displayed as MMDDYY.	&FOCSYSMDYY
&FOCSYSDMY	Current (system clock) date. Displayed as DDMMYY.	&FOCSYSDMYY
&FOCSYSYMD	Current (system clock) date. Displayed as YYMMDD.	&FOCSYSYMD
&MDY	Date that you started the FOCUS session. Displayed as MMDDYY.	&MDYY
&DMY	Date that you started the FOCUS session. Displayed as DDMMYY.	&DMYY
&YMD	Date that you started the FOCUS session. Displayed as YYMMDD.	&YYMD

A Note About Smart Dates

IBI recommends that you use Smart Dates for storing all date values. Smart Dates are machine-readable only. FOCUS stores an integer that represents an offset from the system base date (for example, December 31, 1900, the last day of the 19th century, is Day 0, which means January 1, 1901 is Day 1). Dates that fall before Day 0 are stored as negative integers.

What's actually stored for each Smart Date format depends on the lowest level of granularity for each Smart Date as follows:

Format's lowest level of granularity	Stored as	Zero date is	"1" date is
Day (for example, YYMD, DMYY, etc.)	Integer offset	Dec. 31, 1900	Jan. 1, 1901
Month (for example, YYM, MYY, etc.)	Integer offset	January 1901	February 1901
Quarter	Integer offset	January 1901	February 1901

FOCUS Desktop

Year	Actual integer	None	None
------	----------------	------	------

Important: If you use EDA, base dates for quarters and months differ between EDA and FOCUS SIX. FOCUS Desktop Release 7.20 will use the EDA base dates. FOCUS Desktop automatically migrates databases created in FOCUS Six to the new base dates:

Format's lowest level of granularity	Stored as	Zero date	"1" date
Day (for example, YYMD, DMY, etc.)	Integer offset	Dec. 31, 1900	Jan. 1, 1901
Month (for example, YYM, MYY, etc.)	Integer offset	**December 1900	**January 1901
Quarter	Integer offset	**December 1900	**January 1901
Year	Actual integer	None	None

** Note that for monthly and quarterly dates, FOCUS Seven Desktop will use the same base date as EDA, December, 1900. In the same circumstances, FOCUS Six uses a base date of January, 1901, the same the base date as Core FOCUS.

Converting Full Month and Quarter Containers Back to Full Dates

If you convert dates between month/quarter formats and day formats, they are resolved as follows:

Date format specified...	Resolves to...
A month and a full 4-digit year	First day of the month, so M1/1999 = 01/01/1999
A quarter and a full 4-digit year	First day of that quarter, so Q1/1999 = 01/01/1999

Converting Databases

The following sections list examples that you can use to make your own databases and applications Year 2000 compliant. These examples are the same sample databases that ship with FOCUS Six.

Converting the EMPDATA Database

Following is a simple example of EMPDATA. This database consists of one date field, HIREDATE, which uses format YMD. Since this is a Smart Date format, you can simply edit the EMPDATA.MAS file, changing the format as follows:

Change the following from

```
FIELDNAME=HIREDATE, ALIAS=HDAT, FORMAT=YMD, $
```

to:

```
FIELDNAME=HIREDATE, ALIAS=HDAT, FORMAT=YYMD, $
```

You do not need to dump and load the data; if you do a query on the HIREDATE field, FOCUS Six release 6.40 will automatically return it from the database in YYMD format (for example, a full date container).

To complete your Year 2000 Remediation, make sure any reports that display this field do not reformat it to a partial date container (for example, display the field as HIREDATE/MDY or HIREDATE/YMD). You will also want to make sure that all data currently in the database uses the correct century. Only after the format change to YMD can dates with a century value of '20' be added.

Converting the EMPLOYEE Database

Since the EMPLOYEE database uses integer dates with a field length of six ("I6"), converting its date values to their Year 2000 compliant equivalents requires more work than the conversion for the EMPDATA database, which uses Smart Dates.

To do this conversion, you need to make a copy of the EMPLOYEE.MAS Master File description with date fields formatted using full date containers (for example I8YYMD or YYMD fields instead of the I6YMD fields in the original Master File).

Next, you need to create a FOCEXEC that:

1. "Dumps" all data segments and records from the EMPLOYEE.FOC database to a flat file.
2. Runs CREATE FILE to create the FOCUS (FOC) file that will be the target for the conversion.
3. Reads the values from the flat file into memory.
4. Performs a COMPUTE using the new DA FUSELIB CY function to resolve the correct century for each six-digit date (for example, the HIREDATE field) to convert the date value to a Smart Date. The algorithm used must correctly adjust for the base date differences between FUSELIB and FOCUS Six (for example, you need to subtract 365 days from the result).
5. Writes the new values out to the correct segments in the resultant FOCUS (FOC) file.
6. Does a TABLE request on both the old and new files so you can compare the results.

Once you run the FOCEXEC and it passes your testing, you can replace the original database and Master File description with the newly created database and Master File description.

Remediation in FOCUS Desktop (FFW7.x)

FOCUS Desktop delivers a comprehensive set of new features that allows existing applications to become "century aware," as well as features designed for ease of new development. These fall into the following categories:

1. Global SET parameters to specify under what conditions to assume the 21st century.
2. File and field level parameters to more precisely indicate the assumed century.
3. New Dialogue Manager variables in extended formats.

In general, FOCUS code that was written with four-digit year formats such as MM/DD/YYYY or YYYYMMDD will already work properly without requiring these features. These new features are designed to simplify the conversion of two-digit year display formats to the proper internal date.

FOCUS allows century processing against virtual fields created by DEFINE and COMPUTE statements. A DEFINE is allowed to be coded in a Master File description, and century processing attributes can also be applied in this context. At the virtual field level, two Master File description parameters (DEFCENT and YRTHRESH) provide a means of interpreting the century if the first two digits of the year (YYyy) are not provided elsewhere. If the first two digits are provided, they are simply accepted and validated.

Virtual field level usage of these parameters overrides and takes precedence over any application, file, or field level setting as described in *Describing Data Files*, but only for the virtual field in which they are used. This means that applications with unusual structures can still exert global control over the processing of century data.

The syntax for COMPUTE and DEFINE is:

```
field/format [{DEFCENT|DFC} cc {YRTHRESH|YRT} yy] [MISSING...] = <exp>;
```

where:

`field`

Is the virtual field name being created.

`format`

Is date data type for the virtual field being created.

`cc`

Is the default century value to be applied.

`yy`

Is the threshold value used for application of the century value.

`exp`

Is the logic, routine, or constant being evaluated for the virtual field.

Note:

- When DEFCEMENT is used in combination with YRTHRESH, it establishes a base year for a 100-year span. Any two-digit year is assumed to fall within that span, and the first two digits are set accordingly.
- The Default Century and Year Threshold values are applied to the final result of the right-hand side of an expression (settings that individual fields within an expression may have are ignored). Thus, complex expressions in which this is not the desired behavior should use the technology to spontaneously convert the fields to virtual Smart Date fields, and then perform computation using the Smart Date versions of the fields.
- The DEFCEMENT and YRTHRESH parameters may be abbreviated to DFC and YRT, respectively.

Using the new parameters, you can have FOCUS provide the first two digits in the following cases:

- Conversion from old dates to Smart Dates when the first two digits are not provided.
- Conversion from ACTUAL to USAGE when the first two digits are not provided.
- Data entry when the first two digits are not provided.

Automatic validation will only be done for Smart Dates, ensuring that the date provided is a valid date. Century specification is implemented using the SET parameters, DEFCEMENT and YRTHRESH.

The syntax of DEFCEMENT is:

```
SET DEFCEMENT = nn
```

where:

nn

Is 19 unless otherwise specified.

The syntax of YRTHRESH is:

```
SET YRTHRESH = nn
```

where:

nn

Is zero unless otherwise specified.

The combination of DEFCEMENT and YRTHRESH establishes a base year for a 100-year window. Any two-digit year is assumed to fall within that window, and the first two digits are set accordingly. Years outside the declared window must be handled by user coding.

The default values for the two parameters are DEFCEMENT= 19, YRTHRESH= 00. When you provide a year threshold, years greater than or equal to that value assume the century value assigned by DEFCEMENT. Years lower than that threshold assume a century value of DEFCEMENT plus 1.

To see how DEFCEMENT and YRTHRESH are applied to interpret two-digits years, consider the following:

FOCUS Desktop

```
DEFCENT=19, YRTHRESH=80
```

This describes a range from 1980 to 2079. If a two-digit year field contains 99, the year is 1999. If it's 79, the year is 2079. If it's 00, the year is 2000.

Four new settings were added for Master File descriptions, two each for specifying file and field level handling. Settings made at the field level take precedence over file level settings, which in turn take precedence over global settings. DEFINES and COMPUTEs are supported.

At the file level, the settings FDEFCENT and FYRTHRESH were added. The FDEFCENT syntax is:

```
FDEFCENT
FDFC      = nn
```

where:

```
nn
    Is 19 unless otherwise specified.
```

The FYRTHRESH syntax is:

```
FYRTHRESH
FYRT      = nn
```

where:

```
nn
    Is zero unless otherwise specified.
```

At the field level, DEFCENT and YRTHRESH have been added. The syntax is:

```
DEFCENT
DFC       = nn
```

where:

```
nn
    Is 19 unless otherwise specified.
```

The syntax for YRTHRESH is:

```
YRTHRESH
YRT       = nn
```

where:

```
nn
    Is zero unless otherwise specified.
```

The syntax for COMPUTE and DEFINE is:

```
DEFINE FILE EMPLOYEE
    fld/fmt  [{DEFCENT|DFC} nn {YRTHRESH|YRT} nn] [MISSING...] = <exp>;
...

```

END

The DFC and YRT syntax must follow the field format information.

For example:

```
FILENAME=EMPLOYEE, SUFFIX=FOC, FDEFCENT=20, FYRTHRESH=66, $SEGMNAME=EMPINFO,
SEGTYPE=S1 FIELDNAME=EMP_ID, ALIAS=EID,

FORMAT=A9, $FIELDNAME=LAST_NAME, ALIAS=LN, FORMAT=A15, $FIELDNAME=FIRST_NAME,
ALIAS=FN, FORMAT=A10,

$FIELDNAME=HIRE_DATE, ALIAS=HDT, FORMAT=I6YMD, DEFCENT=19, YRTHRESH=75, $
```

To see the application of DEFCENT and YRTHRESH interpret two-digit years, consider the following:

```
DEFCENT=19, YRTHRESH=80
```

This describes a range from 1980 to 2079. If a two-digit year field contains 99, the year is 1999. If it's 79, the year is 2079. If it's 00, the year is 2000.

The following two examples illustrate the conversion of a two-digit year field with DEFINE and COMPUTE, respectively:

```
DEFINE FILE EMPLOYEE
ESHIRE_DATE/YYMD = HIRE_DATE;
ESHIRE DFC 19 YRT 80 = HIRE_DATE;
END
```

In the example above, the format of HIRE_DATE is I6YMD.

```
TABLE FILE EMPLOYEE
SUM SALARY AND COMPUTE
ESHIRE_DATE/YYMD = HIRE_DATE;
END
```

In the example above, YYMD uses whatever is specified.

Testing

After changes have been made to your application to render it Year 2000 compliant, it's important to test it. The testing of FOCUS for Windows included the following key dates:

12/31/1998

09/09/1999

12/31/1999 (last day before date rollover)

01/01/2000 (first day after rollover)

02/28/2000 (day before leap day 2000)

FOCUS Desktop

02/29/2000

03/01/2000

12/31/2000

01/01/2001

02/28/2001

Your applications should also be tested using this data to make sure that they are Year 2000 compliant.

10 Maintain

Year 2000 Dates and Maintain

As of release 7.08R, Maintain respects global, master and field level settings of YRTHRESH and DEFCENT. These settings work for any database field. A computed field in an application does not respect global settings. If you wish to create a date field with Year 2000 settings, do the following:

```
COMPUTE DATE1/YYMD DFC 19 YRT 10 = Value;
```

Here, value can be any format. It can be a field with an alpha or old date format:

```
COMPUTE DATE2/A6 = '980101';  
COMPUTE DATE1/YYMD DFC 19 YRT 10 = DATE2;
```

Maintain does not require any date subroutines. You can add days, months or years to a date just by doing the arithmetic. You can also determine the number of days between two years just by subtracting one from the other. For examples, see the article "Maintain Makes Date Math Easy," by Mark Derwin, in the special Year 2000 issue of *Information Builders Systems Journal*, September /October 1998, Volume 11, Number 4 and 5, pg. 156. Maintain also allows you to convert formats without needing subroutines or EDIT functions. You can equate an alpha field to a date field. If the alpha field contains a valid date, the value of the data field is automatically set.

Getting Today's Date

While you can use the TODAY function to retrieve the system date, Maintain does not support the return of the century. Using the above COMPUTE technique makes it easier to get the century value. The code would be:

```
MAINTAIN  
COMPUTE X/A8=TODAY(X);  
COMPUTE Y/MDYY=X;  
TYPE "<<Y"  
END
```

The output is today's date with the century.

Maintain

New TODAY and TODAY2 subroutines are available after Release 7.08R. TODAY2 can be used to return the century. The syntax for these routines slightly differs from the syntax for the old TODAY and TODAY2 routines. You do not need to pass a parameter to either of them. Also, you must include the MODULE IMPORT command in your code. The syntax is:

```
MAINTAIN
MODULE IMPORT (MNTUWS);
COMPUTE XYZ/A8      =TODAY();
COMPUTE XYZ1/A10=TODAY2();
TYPE "<<XYZ <<XYZ1"
END
```

The output contains two dates. The second date has the century, but the first one does not.

Note: The TODAY2 function does not support the TESTDATE setting.

Converting Data With Maintain

If you have a Master File that contains an old date field, you can use Maintain to change the format of the field and the data. There are two scenarios that we will look at. In both cases, we will create a duplicate Master File description. In the first case, the two Master Files have identical field names. In the second scenario, we will rename the date field.

Scenario 1:

For these examples, we are going to use the Movies sample file. However, the format of the date field RELDATE has been changed to A6YMD. It does not matter what the format is. It could be an I6YMD, an A6 or any other format used for storing dates.

The Master File description for the Movies file is:

```
FILENAME=MOVIES, SUFFIX=FOC
SEGNAME=MOVINFO, SEGTYPE=S1
FIELDNAME=MOVIECODE, ALIAS=MCOD, FORMAT=A6, INDEX=I, $
FIELDNAME=TITLE, ALIAS=MTL, FORMAT=A39, $
FIELDNAME=CATEGORY, ALIAS=CLASS, FORMAT=A8, $
FIELDNAME=DIRECTOR, ALIAS=DIR, FORMAT=A17, $
FIELDNAME=RATING, ALIAS=RTG, FORMAT=A4, $
FIELDNAME=RELDATE, ALIAS=RDAT, FORMAT=A6YMD, $
FIELDNAME=WHOLESALEPR, ALIAS=WPRC, FORMAT=F6.2, $
FIELDNAME=LISTPR, ALIAS=LPRC, FORMAT=F6.2, $
FIELDNAME=COPIES, ALIAS=NOC, FORMAT=I3, $
```

The Master File description for the Movies2 file is:

```

FILENAME=MOVIES2,      SUFFIX=FOC
SEGNAME=MOVINFO,      SEGTYPE=S1
  FIELDNAME=MOVIECODE,      ALIAS=MCOD,      FORMAT=A6, INDEX=I,$
  FIELDNAME=TITLE,          ALIAS=MTL,          FORMAT=A39,
  FIELDNAME=CATEGORY,      ALIAS=CLASS,      FORMAT=A8,      $
  FIELDNAME=DIRECTOR,      ALIAS=DIR,          FORMAT=A17,      $
  FIELDNAME=RATING,        ALIAS=RTG,          FORMAT=A4,        $
  FIELDNAME=RELDATE,       ALIAS=RDAT,        FORMAT=YMD,        $
  FIELDNAME=WHOLESALEPR,   ALIAS=WPRC,        FORMAT=F6.2,      $
  FIELDNAME=LISTPR,        ALIAS=LPRC,        FORMAT=F6.2,      $
  FIELDNAME=COPIES,        ALIAS=NOC,          FORMAT=I3,        $

```

The two Master Files, Movies and Movies2, are the same except for the format of the RELDATE field. The following code reads all the records from the Movies file, and stores them in MOVSTK. We also create a stack call MOVSTK2 that references the Movies2 file.

We cannot just copy the data from one stack to the other, because their date fields have different formats. Instead, we perform a loop and equate all of the fields and values from MOVSTK into MOVSTK2. Once that stack is populated, we can save it to Movies2.

```

MAINTAIN FILE MOVIES AND MOVIES2
INFER MOVIES2.MOVIECODE INTO MOVSTK2
FOR ALL NEXT MOVIES.MOVIECODE INTO MOVSTK
  COMPUTE J/I2=1;
  REPEAT MOVSTK.FOCCOUNT
    COMPUTE MOVSTK2(J).MOVIECODE = MOVSTK(J).MOVIECODE;
    COMPUTE MOVSTK2(J).TITLE = MOVSTK(J).TITLE;
    COMPUTE MOVSTK2(J).CATEGORY = MOVSTK(J).CATEGORY;
    COMPUTE MOVSTK2(J).DIRECTOR = MOVSTK(J).DIRECTOR;
    COMPUTE MOVSTK2(J).RATING = MOVSTK(J).RATING;
    COMPUTE MOVSTK2(J).RELDATE = MOVSTK(J).RELDATE;
    COMPUTE MOVSTK2(J).WHOLESALEPR= MOVSTK(J).WHOLESALEPR;
    COMPUTE MOVSTK2(J).LISTPR = MOVSTK(J).LISTPR;
    COMPUTE MOVSTK2(J).COPIES = MOVSTK(J).COPIES;
    COMPUTE J=J+1;
  ENDREPEAT
FOR ALL INCLUDE MOVIES2.MOVIECODE FROM MOVSTK2
END

```

At the end of this procedure, the Movies2 file contains all the data from the Movies file, but the RELDATE field is now a Smart Date.

Maintain

Scenario 2:

In the second technique, we change the format and the name of the RELDATE field. In Movies3, we rename RELDATE to RELDATE1, and change the format to a YYMD field.

The Movies3 Master File description is:

```
FILENAME=MOVIES,      SUFFIX=FOC
SEGNAME=MOVINFO,     SEGTYPE=S1
FIELDNAME=MOVIECODE, ALIAS=MCOD,      FORMAT=A6, INDEX=I, $
FIELDNAME=TITLE,     ALIAS=MTL,       FORMAT=A39,      $
FIELDNAME=CATEGORY,  ALIAS=CLASS,     FORMAT=A8,       $
FIELDNAME=DIRECTOR,  ALIAS=DIR,       FORMAT=A17,     $
FIELDNAME=RATING,    ALIAS=RTG,       FORMAT=A4,       $
FIELDNAME=RELDATE1,  ALIAS=RDAT,     FORMAT=YYMD,    $
FIELDNAME=WHOLESALEPR, ALIAS=WPRC,     FORMAT=F6.2,    $
FIELDNAME=LISTPR,    ALIAS=LPRC,     FORMAT=F6.2,    $
FIELDNAME=COPIES,    ALIAS=NOC,       FORMAT=I3,      $
```

The following code reads all the data from the Movies file into MOVSTK1. We then add an additional column to the stack that represents the new date field in the Movies3 file. We perform a loop and populate that field with the data in the Movies file. Finally, we include all the data in the Movies file.

```
MAINTAIN FILE MOVIES AND MOVIES3
FOR ALL NEXT MOVIES.MOVIECODE INTO MOVSTK1
COMPUTE I/I3=1;
REPEAT MOVSTK1.FOCCOUNT
  COMPUTE MOVSTK1(I).RELDATE1/YYMD=MOVSTK1(I).RELDATE;
  COMPUTE I=I+1;
ENDREPEAT
FOR ALL INCLUDE MOVIES3.MOVIECODE FROM MOVSTK1
END
```

This method has fewer lines, but requires you to rename the date field.

Maintain's ability to easily perform data format conversions makes it the perfect candidate for Year 2000 solutions.

11 SmartMode/SiteAnalyzer

Year 2000 Capable Releases and Enhancements

SmartMode Release 7.0.8R for FOCUS and Release 3.2.3 for EDA are considered Year 2000 capable, as are SiteAnalyzer Release 7.0.8R for FOCUS and Release 3.2.3 for EDA. The metadata descriptions for all SmartMode and SiteAnalyzer for FOCUS internal tables have been updated to reflect four-digit years. In addition, all procedures and reports that reference date columns have been updated to use four-digit years.

SiteAnalyzer Release 7.0.7 and above for FOCUS include valuable Year 2000 reporting tools which can help identify date fields within FOCUS Master File descriptions and procedures. This functionality is not available in SiteAnalyzer for EDA.

Non-Year 2000 Capable Releases

Releases of SmartMode and SiteAnalyzer prior to Release 7.0.8R for FOCUS are not certified as Year 2000 capable, since the dates in the internal tables and reports for SmartMode and SiteAnalyzer are not century aware. Releases of SmartMode and SiteAnalyzer for EDA prior to Release 3.2.3 are not certified as Year 2000 capable, as support of this environment is limited to certified EDA releases. Support, in this context, refers to the availability of new software fixes. For this reason, Information Builders recommends that you upgrade to the Year 2000 capable releases of these products.

SmartMode/SiteAnalyzer

12 Year 2000-Related Known Problems

The following known problems are either related to Year 2000 issues or are date issues that you may come across during remediation. The technical support section of our www.ibi.com web site has a more complete and up to date database of known problems that are available to our InfoResponse Online customers.

All known problems are listed by product and then case number.

Case Number: 40051007 Date Reported: 99/01/05

Summary: MNTN:YM & MYY dates return incorrect results(1 mnth forward)

Status: Submitted to Product Division

Product: Cactus:Cross-Platform Maintain Op Sys: WIN Major Release: 6.4

Description:

Dates with format YM and MYY returns incorrect month data, when used with MAINTAIN. Example : 03/1998 is returned as 04/1998. TABLE returns correct data.

Problem appears in CACTUS 3.7 dated 10-15-98, FFW 6.40.

Appears to be fixed in CACTUS dated 12-14-98, FFW 6.40b, but only for data that is entered using the 6.40b version.

Workaround:

None.

Year 2000-Related Known Problems

Case Number: 31261141 Date Reported: 98/05/06

Summary: Copymanager:Recurring schedule requires stop date in 3.3.1

Status: Closed

Product: Copy Manager Op Sys: WIN/NT Major Release: 331

Description:

In the pre-3.3.1 Copy Manager client, you could schedule a copy request as recurring and it would allow the STOP DATE/TIME to be omitted. The new 3.3.1 Copy Manager Client requires an entry for STOP DATE/TIME. This prevents you from easily scheduling a recurring request to run forever. This entry should not be necessary.

Workaround:

Use end date of 12/31/99 in the STOP DATE/TIME entry.

Case Number: 32121093 Date Reported: 98/07/31

Summary: COPY MANAGER: COPY LOG SHOWS DATE 12/31/1900 FOR -12 ERRORS

Status: Submitted to Product Division

Product: Copy Manager Op Sys: AIX Major Release: 331

Description:

The following changes are requested for Copy Manager:

1. Change the Copy Client to allow it to accept only 8 characters for an

Year 2000-Related Known Problems

RPC name.

2. Fix the datestamp in cmlog2 as when a -12 error occurs, it shows a date of 12/31/1900 when the system date had been reset for year 2000.

The date value of 12/31/1900 is always the one logged in the copy log even though the copy request ran on "system dates" of 01/01/2000, 02/28/2000, 02/29/2000 and 01/02/2001.

For example:

```
1900/12/31 03.14.17 rkvmora1 (FOC35804) BAD STATUS CODE (-12) RECEIVED
FROM
SERVER CMJOB;90211622
1900/12/31 03.14.17 rkvmora1 (FOC35862) EDA STATUS (-12): ERROR RECEIVING
DAT
FROM SERVER;90211622
1900/12/31 03.14.17 rkvmora1 (FOC1400) SQLCODE IS -12 (HEX: FFFFFFF4);90211622
```

Workaround:

None.

Case Number: 31241003 Date Reported: 98/05/04
Summary: -TYPE &DATEYYMD returns 2010/2011 for Year 2000 dates
Status: Submitted to Product Division
Product: EDA Op Sys: WIN/NT Major Release: 3

Description:

-TYPE &DATEYYMD returns the year 2010 wrong system date for dates in year 2000.

Year 2000-Related Known Problems

The left column below is the actual date, the right column shows the incorrect date displayed via &DATEYYMD:

2001 --> 2010

2002 --> 2010

...

2009 --> 2010 also anything over 2011 generated a date of 2011

Workaround:

Patch I0024 corrects this problem. The correct system dates are returned once this patch is applied. The patch is a DLL.

This patch can be obtained from customer support.

Case Number: 31951011 Date Reported: 98/07/14

Summary: CURRENT TIME translated incorrectly-returns ORACLE 904 err

Status: Submitted to Product Division

Product: EDA Op Sys: UNIX Major Release: 3.2

Description:

When using current time, EDA returns an Oracle error 904. The tracing shows a TO_TIME function being translated. This function is not valid for Oracle .

For example:

```
select current time from test2000;
```

is translated into

```
SELECT TO_TIME ('14:58:28' ) FROM TEST2000;
```

Year 2000-Related Known Problems

Oracle returns a 904 error meaning unknown column.

Workaround:

none

Case Number: 32041062 Date Reported: 98/07/23

Summary: YEAR2000:IBITRACE shows year 1900 not 2000 w/tracestamp ON

Status: Closed

Product: EDA Op Sys: AIX Major Release: 33

Description:

The trace file, FSTRACE.FTM, created by the IBITRACE utility is displaying system date of 1900 when the system date has been set for a year 2000 date when tracestamp is set on.

```
SET TRACEON=ALL//fstrace
SET TRACEOFF=stdoutx
SET TRACESTAMP = RIGHT
```

For a system date of 2000-01-31 it displays 1900-01-31.

Workaround:

This problem will be addressed in the EDA 4.x server.

Year 2000-Related Known Problems

Case Number: 32261114 Date Reported: 98/08/14
Summary: FOC1400(-1 sqlcode) and 1406 with IF test on RDB date range
Status: Submitted to Product Division
Product: EDA Op Sys: VMS Major Release: 331

Description:

The SQL translator generates invalid Rdb SQL for a date range test.
The following IF condition in a TABLE request generates the following SQL:

IF LOGIN_DATE FROM '19980701' TO '19980701'

WHERE (T1.LOGIN_DATE BETWEEN
TO_DATE('01-07-1998','DD-MM-YYYY') AND TO_DATE('01-07-1998','DD-MM-
YYYY'))

TO_DATE is actually an ORACLE function, which is why this fails against RDB.

This generates the following errors:
(FOC1400) SQLCODE IS -1 (HEX: FFFFFFFF)
: %SQL-F-RTNNOTDEF, function or procedure !AC is not deined
(FOC1406) SQL OPEN CURSON ERROR. :CSAGTACS

Workaround:

None.

Case Number: 30891062 Date Reported: 98/03/30

Year 2000-Related Known Problems

Summary: YEAR2000:EDACFG in 331 displays 2010 for all year2000 dates

Status: Submitted to Product Division

Product: EDA Server Op Sys: AIX Major Release: 331

Description:

There is a date routine used by edacfg which will not properly identify the four digit year after 1999. When edacfg is run to configure an EDA server on an AIX 4.1.5 system that has its system date set later than 12/31/1999, the value that is displayed for year is 2010. For example, 01/01/2000 is displayed as 01/01/2010 and 01/01/2001 is displayed as 01/01/2010.

This problem also affects Copy Manager as the year stored in the scheduler database and displayed on reports is also 2010 for year 2000 dates. In addition a leap year date value of 02/29/2000 was said to be an invalid date by Copy Client/Server.

Here is the output from edacfg:

SYSPRINT.SCR

(FOC1400) SQLCODE IS -1 (HEX: FFFFFFFF)

(FOC1488) SQL INTERFACE IS NOT INSTALLED

: Physical interface not found for SUFFIX=SQLINF . Module name:

INFFOC

0 NEW FILE SYSCOLLN ON 01/04/2010 AT 01.57.47

0 NEW FILE SYSCOLLT ON 01/04/2010 AT 01.57.47

0 NEW FILE SYSRPC ON 01/04/2010 AT 01.57.47

0 NEW FILE FOCUSSU ON 01/04/2010 AT 01.57.47

0 SYSCOLLN ON 01/04/2010 AT 01.57.47

Year 2000-Related Known Problems

```
0 TRANSACTIONS:  TOTAL =  3  ACCEPTED=  3  REJECTED=  0
  SEGMENTS:      INPUT =  3  UPDATED =  0  DELETED =  0
0 SYSCOLLT      ON 01/04/2010 AT 01.57.47
0 TRANSACTIONS:  TOTAL = 14  ACCEPTED= 14  REJECTED=  0
  SEGMENTS:      INPUT = 14  UPDATED =  0  DELETED =  0
0 SYSRPC        ON 01/04/2010 AT 01.57.47
0 TRANSACTIONS:  TOTAL =  2  ACCEPTED=  2  REJECTED=  0
  SEGMENTS:      INPUT =  2  UPDATED =  0  DELETED =  0
0 SYSRPC        ON 01/04/2010 AT 01.57.48
0 TRANSACTIONS:  TOTAL =  9  ACCEPTED=  9  REJECTED=  0
  SEGMENTS:      INPUT =  9  UPDATED =  0  DELETED =  0
CLOSEALL.ALL
%disconnect
%stop_server
```

```
/3rd_lpp/eda/qry331/bin>date
Tue Jan  4 01:59:23 PST 2000
/3rd_lpp/eda/qry331/bin>
```

Workaround:

Patches I0043, I0044, I0045 & I0047 address this problem.

Case Number: 20301052 Date Reported: 97/01/30
Summary: DEFINE:DATE: Problem with FROM TO range and P17.2
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 706

Description:

Using a FROM TO DATE in DEFINE with the Date on the FROM Range in conjunction with P17.2 yields incorrect output. If the define is changed to P16.2 it works. Also, if the WHERE test is going against a real field with a format of P17.2 it is okay. The problem only occurs with a DEFINED field and a date range test.

Workaround:

Change the DEFINED field to P16.2 format instead of using a P17.2 format.

Case Number: 21051053 Date Reported: 97/04/15

Summary: Y2K:YM or YYM cannot differentiate between 0 and 1901/01

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 70

Description:

FOCUS cannot differentiate between a 0 and the value 1901/01 with the format of YM or YYM. When the field is printed, it shows up as a blank.

Workaround:

N/A

Year 2000-Related Known Problems

Case Number: 21201113 Date Reported: 97/04/30
Summary: WHERE field is Jan 30, 1899 does not work on YMD format.
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 706

Description:

A 'WHERE' test accessing dates in a FOCUS file, where the dates span centuries will return the wrong data. For example, WHERE field is Jan 30, 1899 will look for records that is equal to Jan 30, 1999. The default century is used instead of what was explicitly specified in the WHERE test.

Workaround:

An 'IF' test will produce the correct output.
This is fixed by the Mega Date PTF for release 7.0.8 and is included in release 7.0.8R.

Case Number: 21431050 Date Reported: 97/05/23
Summary: In 7.0.7M, MFD with ACTUAL=A8YYMD USAGE=YYMD gets FOC1346
Status: Closed
Product: FOCUS Op Sys: VM Major Release: 707

Description:

A FOC1346 format error is generated when a date field in an external file that has an ACTUAL of A8YYMD and a USAGE of YYMD and contains zeroes in the incoming

field.

Workaround:

Change the ACTUAL from A8YYMD to A8.

Case Number: 21751098 Date Reported: 97/06/24

Summary: *NFR* ACTUAL of A6YMD, USAGE YMD gets FOC1130 or FOC200

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 707

Description:

Customers need the ability to identify MISSING dates, where the USAGE is a "smart date", (MDY, YMD, YYMD, DMY, etc.), and the ACTUAL is either A6 or I6.

Currently, if a field is stored, with a USAGE = YMD (or MDY, etc), MISSING=ON, FOCUS will recognize that the field is MISSING only under one of two scenarios:

ACTUAL = A6 value is . (a period)

ACTUAL = DATE value is -9998998

Any other non-date values will give a FOC200 or FOC1130 and ultimately a blank value (default date of 12/31/00) on the report.

NEW FEATURE REQUEST:

For EXTERNAL files, when MISSING = ON is coded for a field with a format of a "smart date", provide an additional parameter MISSVAL to indicate the actual value which identifies MISSING. This would follow the attribute that is currently available in ANALYSE. If MISSING is specified, but MISSVAL is not, then the FOCUS default of . or -9998998 would identify the field as MISSING.

Year 2000-Related Known Problems

Workaround:

N/A

Case Number: 22791109 Date Reported: 97/10/06

Summary: FTOA subroutine is zeroing out DEFCENT and YRTHRESH

Status: Closed

Product: FOCUS Op Sys: MVS Major Release: 707

Description:

The FTOA subroutine appears to SET DEFCENT and YRTHRESH to 0. This only occurs when PTF #71834 from PROJECT #23928 is applied to 7.0.7M.

It does not occur on any other release of FOCUS.

Workaround:

This only occurs in 7.0.7M with PTF 71834 applied. This works in rel 7.0.8 and up.

Case Number: 22801085 Date Reported: 97/10/07

Summary: Startdate for SMARTDATES on MF is different from UNIX

Status: Submitted to Product Division

Product: FOCUS Op Sys: SOLARS Major Release: 68

Year 2000-Related Known Problems

Description:

The startdate for SMARTDATES on MF MVS and VM FOCUS is different than the startdate on the UNIX environment. The IMPORT facility does NOT translate the date correctly.

Workaround:

Use FIXFORM to load the data from MF to UNIX

Case Number: 23421123 Date Reported: 97/12/08

Summary: UC:707-708:REBUILD incorrect with selection on smart date

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM/ESA Major Release: 708

Description:

REBUILD/REORG DUMP Does not work correctly if there is a record selection test on a "smartdate" field.

If the selection criteria is "GE" or "GT", all records are dumped.

If the selection criteria is "LE" "LT" or "EQ", no records are dumped for the segment containing the smart date field tested.

Workaround:

This is an upward compatibility issue and there is no workaround.

This is fixed by PTF 79580 in 7.0.8.

Year 2000-Related Known Problems

Case Number: 23521031 Date Reported: 97/12/18

Summary: CHGDAT does not display '000229' correctly moving to YYMD fmt

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 708

Description:

CHGDAT subroutine does not recognize the date value '000229' when DEFCENT and YRTHRESH are set and the conversion is from 'YMD' to 'YYMD' format. It works correctly when going to any format other than YYMD.

Workaround:

Use a different format, or include the century and pass the subroutine an eight digit date.

Case Number: 23561067 Date Reported: 97/12/22

Summary: Run FEX verifying Y2K dates: SOC4 ABEND: worked fine in 707M /

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

The following DEFINE statement:

NEWFIELD/A10='&DATEYYMD'; --- will cause an OC4 abend in MVS and a DMS

Year 2000-Related Known Problems

abend in CMS if the DEFINE statement is NOT the last DEFINE for a particular file.

Workaround:

Placing the statement as the last of the DEFINES prevents the abend.

Case Number: 30121088 Date Reported: 98/01/12

Summary: DOWK subroutine does not recognize DEFCENT/YRTHRESH Settings

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 7.0

Description:

The DOWK subroutine does not recognize the DEFCENT and YRTHRESH settings when used with 6-digit, I6YMD fields. The correct day of the week is NOT returned. This also occurs in Dialogue Manager.

Workaround:

While this is not a workaround for 6-digit incoming fields, 8-digit, I8YYMD fields will produce the correct day of the week.

This is fixed by the Mega Date PTF for 7.0.8 and also included in 7.0.8R.

Case Number: 30161028 Date Reported: 98/01/16

Summary: DATEFNS=ON and nested EDITS within GREGDT subroutine: FOC244

Year 2000-Related Known Problems

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

Nested EDITs within the GREGDT subroutine causes a FOC244 when DATEFNS is set to 'ON'.

Workaround:

Do the EDITs outside of the GREGDT subroutine.

Case Number: 30201024 Date Reported: 98/01/20

Summary: UC: YY = 0 does not produce same results.

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 700

Description:

Entering a value of 0 or 0000 for a YY field results in a (FOC177) INVALID DATE CONSTANT: ;

Page 9-78 in the 7.0 users manual states a 0 value can be entered for a date and should be interpreted as a blank. This used to work, and is broken in release 7.0.8 (700-9609).

Workaround:

DATE_FIELD/YY = ;

Case Number: 30281102 Date Reported: 98/01/28
Summary: 7.0.8:Dialogue Manager with CHGDAT gives blanks
Status: Closed
Product: FOCUS Op Sys: MVS Major Release: 7.0

Description:

CHGDAT subroutine requires two arguments specifying input and output formats. If those format specifications are not supplied with 5 characters, the results of the subroutine may be blank. This is especially true if a previous call to the subroutine specified a format with more than the subsequent formats. This is a problem in FOCUS 7.08 with or without the mega PTF in both MVS and CMS operating systems. Works in FOCUS 7.0.8R.

Workaround:

Add enough characters in arguments 1 and 2 to pad out to 5 places, ie.
-SET &DATE_1 = CHGDAT(YYMD.,MDYYX,&TEMP_DATE_1 | 'X', 'A17');
-SET &DATE_2 = CHGDAT('YMD ',MDYYX,&TEMP_DATE_5 | 'X', 'A17');
This appears to be fixed in FOCUS 7.08R.

Case Number: 30371062 Date Reported: 98/02/06
Summary: SET &VARIABLE YY=0;, gets FOC177:INVALID DATE CONSTANT

Year 2000-Related Known Problems

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

Setting a 'YY' format field to zeroes(0) will produce a FOC177 error.

Workaround:

This is fixed in release 7.0.8R.

Case Number: 30371125 Date Reported: 98/02/06

Summary: Alternate MFD w/FDEFCENT and FYRTHRESH:Y2K date incorrect

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 707

Description:

A DEFINE is reformatting a previously DEFINED field that was a YMD format to a YYMD format. The alternate master contains FDEFCENT/FYRTHRESH. The DEFINE is not recognizing these settings. When doing the same exact procedure, but going against the primary MASTER, this works. Also, if the DEFINE is based on a real field, not a previously DEFINED field, it also works.

Workaround:

Code an explicit SET DEFCENT/YRTHRESH in the FOCEXEC, prior to the DEFINE FILE, when using an alternate master.

Year 2000-Related Known Problems

Case Number: 30441037 Date Reported: 98/02/13
Summary: Setting HIPERULE to a Year 2000 date causes OC4 abend
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 707

Description:

Using Hiperules for setting HiperFOCUS and testing using Hourglass to set the system date to year 2000 causes an OC4 abend. In this case, the HIPERRULE statement looked like this:

ACCEPT DAY = SATURDAY OR DAY=SUNDAY

When the system date is set to year 2000 and saturday, HiperFOCUS cannot be brought up. When tested with Jan 16th (SUN), it caused an OC4. All other days of the week also abends. This only affects year 2000. Year 2001 and all other years work fine.

Workaround:

A patch is available for this problem for FOCUS 7009605. This problem apparently exists in MF FOCUS as well as EDA when Hiperrule is used. The fix will be included in EDA 4.2.1.

Case Number: 30471032 Date Reported: 98/02/16
Summary: DATEFNS=ON causes performance degradation
Status: Submitted to Product Division

Year 2000-Related Known Problems

Product: FOCUS Op Sys: VM Major Release: 708

Description:

Executing a procedure, which calls date subroutines, with DATEFNS set to 'ON' causes performance degradation. With DATEFNS set to 'OFF' fewer CPU cycles are used, and the procedure runs faster.

Workaround:

N/A

Case Number: 30621115 Date Reported: 98/03/03

Summary: ACE: Need to make Y2K compliant

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 700

Description:

ACE: Not Y2K compliant

Workaround:

N/A

Case Number: 30651025 Date Reported: 98/03/06

Year 2000-Related Known Problems

Summary: DATEFNS /AYM Subroutine output different in 7.0.8, 7.0.7M

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

The AYM subroutine produces different output with DATEFNS set to OFF in 7.0.8, and without using the PROJECT 2000 DEFCENT/YRTHRESH settings in 7.0.7M and prior releases.

Workaround:

N/A

Case Number: 30771023 Date Reported: 98/03/18

Summary: FOC244:When Nested EDIT within YMD functions

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 708

Description:

The use of the EDIT function within the parmater sequence of the YMD function causes a FOC244 error. This error does not occur in releases prior to 7.0.8

Workaround:

Use the EDIT function outside of the calling sequence fo the YMD function.

Year 2000-Related Known Problems

PTF 84147 is now available, which will permit the nested EDIT.

Case Number: 30831088 Date Reported: 98/03/24

Summary: EMR/FRL:Using SMARTDATE:FOR cmd doesn't work

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

A "smart" date cannot be used as the 'FOR' field in an EMR request. This does not give an error message and 0 (zero) lines of output are produced. This occurs in all 7.0.x and 6.8.x releases.

Workaround:

ReDEFINE the YMD field into a I6YMD format.

This is also addressed by the Mega Date PTF, and in 7.0.8R

Case Number: 30901057 Date Reported: 98/03/31

Summary: &DATEMDYY w/ screen attributes in a -CRTFORM gets a FOC295

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 707

Description:

Year 2000-Related Known Problems

The use of the &DATEYYMD variable in a -CRTFORM, using extended screen attributes produces a FOC295. EXTTERM is set to ON.

Workaround:

SET any meaningless variable to &DATEYYMD, and then date value displays without error.

Case Number: 30921046 Date Reported: 98/04/02

Summary: DFC/YRT does not work correctly in MODIFY with COMPUTE

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 7.0

Description:

The use of DEFCENT/YRTHESH in a MODIFY/COMPUTE does not work. An I6YMD field

is converted to a YYMD field and the DEFCENT/YRTHESH settings are applied to this smart date. A date that should appear in the year 2000, instead displays in the year 1900.

Workaround:

There is no workaround. The use of COMPUTE with DEFCENT/YRTHRESH is even more critical as a MAINTAIN feature, where data validation is done manually via COMPUTE.

This is fixed by the Mega Date PTF and is also included in 7.0.8R.

Year 2000-Related Known Problems

Case Number: 31061126 Date Reported: 98/04/16
Summary: UC to 7.0.8:YMD function w/ nested EDIT causes a FOC279
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Description:

Using a nested EDIT within the YMD function produces a FOC279 error
in rel 7.0.8.

Workaround:

Use the EDIT function outside of the calling sequence of the YMD function.
This is fixed by the Mega Date PTF and in 7.0.8R

Case Number: 31101008 Date Reported: 98/04/20
Summary: TODAY funct not returning 4 digit year like other env's
Status: Closed
Product: FOCUS Op Sys: AIX Major Release: 685

Description:

MF/FOCUS 7.0.8R and VAX/FOCUS 6.9.3, both return the 4 digit year when
'A10' is specified as the format in the TODAY function. UNIX/FOCUS
does not.

Port mainframe code to the UNIX environment that will

Year 2000-Related Known Problems

allow the TODAY function to return the 4 digit year.

Workaround:

None.

Case Number: 31131069 Date Reported: 98/04/23
Summary: &DATEY displays as a blank for the year 2000.
Status: Closed
Product: FOCUS Op Sys: VM/ESA Major Release: 7.8

Description:

System variable,&DATEY, when equal to 00 for year 2000 displays as blanks.

Workaround:

This is addressed in release 7.0.8R as a new feature.
SET DATEDISPLAY = ON
This allows FOCUS to display a 0 instead of a blank when the smart date value is equal to the base date.

Case Number: 31181117 Date Reported: 98/04/28
Summary: Assigning P6YMD to YY/M/D format:Century doesn't show up
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Year 2000-Related Known Problems

Description:

When a P6YMD and I6YMD fields are placed in a HOLD file, and these values are used to update YMD fields in a FOCUS file through FIXFORM FROM HOLD/ DATA ON HOLD, and then these fields are reDEFINED as YYMD field, the first two digits are '00' e.g., the value '980404' is displayed as '00980404'. This works correctly in release 7.0.5 through 7.0.7M

Workaround:

Use HOLD FORMAT ALPHA or ON TABLE SAVE
This is also fixed by the Mega Date PTF and in 7.0.8R

Case Number: 31211049 Date Reported: 98/05/01
Summary: YYJUL format doesn't support 4 digit year, shows only 2 digi
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 7.0

Description:

There is no way to display a FOCUS Smart Date in julian format with a 4 digit year. It always displays 2 digits, showing only the year portion with the century.

Workaround:

Convert the Smart Date to a legacy date and then use the JULDAT subroutine.

Year 2000-Related Known Problems

Case Number: 31331123 Date Reported: 98/05/13
Summary: Graphing Smartdate gets 00/00/00/c
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 706

Description:

*** New Feature Request ***

If you use GRAPH FILE with a smart date, the display does not show

gregorian dates, but the offset with edit options that make offsets appear as dates.

Workaround:

The only workaround is to move the smart date to a YYMD smart date, then define a traditional date with the century, and then graph against the traditional date.

Case Number: 31381113 Date Reported: 98/05/18
Summary: YMD gives SIGNAL 10 when 2nd date is blank /c
Status: Closed
Product: FOCUS Op Sys: SOLARS Major Release: 6.8

Description:

Year 2000-Related Known Problems

The YMD function calculates the difference between two dates in ymd order. If a blank date is supplied, the resulting error is:

Signal 10 (SIGBUS) - "Bus Error" - is usually generated when the trouble trying to map a memory address to the physical location. probably, it signifies a program bug.

Workaround:

Instead of a blank date as input, set it equal to '000000

Case Number: 31481018 Date Reported: 98/05/28
Summary: ?rel with defcent=20 gives wrong value, FOCUS 6.8.1.12177
Status: Closed
Product: FOCUS Op Sys: AIX Major Release: 68

Description:

In AIX FOCUS gen 6.8/5.12177c, ? REL will display the create date as being 01/19/1904 when SET DEFCENT=20 is set.

Workaround:

None.

Year 2000-Related Known Problems

Case Number: 31551158 Date Reported: 98/06/04
Summary: YJUL gives a FOC177 and FOC258 with a value of '00366'
Status: Closed
Product: FOCUS Op Sys: MVS Major Release: 706

Description:

A field with a format of YJUL will give a FOC177 and a FOC258 when the value of the Julian date is 00366. YRTHRESH and DEFCEM are set to 75 and 19, respectively indicating that the value is dealing with Dec 31 2000.

Workaround:

N/A

Case Number: 31631019 Date Reported: 98/06/12
Summary: YMD() doesn't work properly when passed 2000/02/29
Status: Closed
Product: FOCUS Op Sys: AIX Major Release: 6.8

Description:

The YMD function should return the difference between two dates as an integer value. For example YMD(20000201,20000228) returns the value of 27. YMD(20000201,20000229) properly returns the value 28 in gens 68/1.12160c and 68/1.11990c. In 68/5/12177c, the value of 0 is returned.

Year 2000-Related Known Problems

Workaround:

Add another DEFINE field which tests for a zero when 20000229 is used, for example:

```
DEFINE FILE CAR
  D_YMD/I4 = YMD(20000201,20000229);
  DTEST2/I4 = IF D_YMD EQ 0 THEN value ELSE D_YMD;
END
```

Case Number: 31821013 Date Reported: 98/07/01

Summary: DEFINE reusing same fieldname w/CHGDAT and EDIT gets bad o/p

Status: Closed

Product: FOCUS Op Sys: VM Major Release: 708

Description:

Using the CHGDAT subroutine twice in a DEFINE with the same fieldname on the left hand side of the statement will give inaccurate results when edited down to less characters on the first DEFINEd field.

This occurs in 7.0.8.

Workaround:

Use unique fieldnames when using a subroutine more than once.

Year 2000-Related Known Problems

Case Number: 31841019 Date Reported: 98/07/03
Summary: NFR:Need for TODAY function to include the century
Status: Closed
Product: FOCUS Op Sys: SOLARS Major Release: 685

Description:

The TODAY function should return a four digit year. Currently it only returns a two digit year.

Workaround:

None

Case Number: 32021032 Date Reported: 98/07/21
Summary: System date not changed calling FOCUS from LPA w/HOURGLASS
Status: Closed
Product: FOCUS Op Sys: MVS Major Release: 700

Description:

Problem when using HOURGLASS to test Y2K with FOCUS release 7.0.8 under TSO with re-entrant modules installed in the LPA. This works fine in a BATCH JOB. The problem only occurs when calling FOCUS from TSO, then HOURGLASS does not change the system date.

Workaround:

Year 2000-Related Known Problems

Do not invoke FOCUS from LPA.

Case Number: 32021077 Date Reported: 98/07/21

Summary: Y2K: DATE problem with dialogue manager and AYM subroutine

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 7.0

Description:

Dialogue Manager automatically zero suppresses all numeric variables, including the output of YM.

Workaround:

There is a workaround however, it does require a fair bit of code changes.

-SET &VAR2 = AYM(9912,1,'14') + 10000;

-SET &YR = EDIT(&VAR2,'\$99');

Case Number: 32021117 Date Reported: 98/07/21

Summary: VM:FOCUS is not showing the right century set by TIMETRAP

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 7.0

Description:

Year 2000-Related Known Problems

On VM/ESA, FOCUS is not showing the correct century that is being set by TIMETRAP. This is a 3rd party software used for simulating the system date. For example, TIMETRAP is setting the system date to 2/28/2001. When going into FOCUS, the FOCUS banner displays the current date as 2/28/1901.

Workaround:

In 7.0.8R, SET TESTDATE can be used to change the system date for FOCUS applications for testing purposes.

Case Number: 32111107 Date Reported: 98/07/30
Summary: DFC and YRT at field level do not work with Date Subroutines
Status: Closed
Product: FOCUS Op Sys: MVS Major Release: 7.0

Description:

User Written Date subroutines do not use the DEFCENT/YRTHRESH settings when specified at the field level.

Workaround:

The workaround is to specify the YRTHRESH/DEFCENT settings at the Global level(SET).

Case Number: 32221005 Date Reported: 98/08/10

Year 2000-Related Known Problems

Summary: UC:708 DEFINE date w/M format & subtract mnths: wrong output

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 7.0

Description:

Upgrading to 7.0.8, a DEFINEd field that is an I8YYMD is reDEFINEd with a M or MT format. It is then reDEFINEd to subtract "n" number of months from the M formatted field. This results in incorrect output.

For example:

```
DEFINE FILE CAR
MONTHEND/YYMD=19980101;
MTHEND/I8YYMD = MONTHEND ;
MONTH/M = MONTHEND ;
MONTHT/MT = MTHEND ;
MONTH1/MT = MONTH - 1 ;
MONTH2/MT = MONTH - 2 ;
END
TABLE FILE CAR
PRINT MTHEND MONTHT MONTH1 MONTH2
BY COUNTRY NOPRINT
END
```

Note, this example shows this problem for the MONTH-1. MONTH-2 is OK. This worked up to and including release 7.0.7M. It does not work in 7.0.8 and 7.0.8 with the mega PTF applied, as well as, in 7.0.8R and 7.0.9.

Workaround:

N/A

Year 2000-Related Known Problems

Case Number: 32261116 Date Reported: 98/08/14
Summary: WHERE test on MDY w/century supplied gets 0 recs. IF works.
Status: Submitted to Product Division
Product: FOCUS Op Sys: VM Major Release: 707

Description:

A WHERE test on a MDY field uses the DEFCENT and YRTHRESH determination even if the full century is supplied. This occurs in release 7.0.7M and 7.0.8R, but is not a problem in release 7.0.8 with PTF87857 applied. This is similar to a problem with DEFINE which did not take century from the supplied digits, but from DEFCENT and YRTHRESH which was solved in release 7.0.8.

Workaround:

Use an IF statement instead of WHERE for the selection criteria or upgrade to 7.0.8 with PTF 87857.

Case Number: 32381040 Date Reported: 98/08/26
Summary: DATEFNS=ON w/many Date routines in large focexec gets FOC206
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Description:

Year 2000-Related Known Problems

Running a Multi-step Table request with heavy usage of Date subroutines,
date & variables and functions gets:
(FOC206) INSUFFICIENT MEMORY IS AVAILABLE FOR PROGRAM/CACHE

Workaround:

```
SET DATEFNS=OFF
```

```
*****
```

Case Number: 32391119 Date Reported: 98/08/27
Summary: FOC006 with year 00 - 31 used in a compare statement
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 700

Description:

Comparing a date to a FOCUS natural date literal with two digit year value
in the range of 00 - 31 result in FOC006 error. If a 4 digit year is used,
or the 2 digit year used is greater than 31, then problem will not occur.
DEFCENT/YRTHRESH setting have no significant in this case.

For Example:

```
DEFINE FILE CAR  
DATE1/MDY = '&DATE';  
END  
TABLE FILE CAR
```

Year 2000-Related Known Problems

Workaround:

Use a natural date literal with 4 digit year, or use 2 digit year whose value is greater than 31.

Case Number: 32461128 Date Reported: 98/09/03

Summary: YRTHRESH=-n does not work with TESTDATE or 3rd party pkg

Status: Closed

Product: FOCUS Op Sys: VM Major Release: 700

Description:

Using TESTDATE (with year greater than 1999) and YRTHRESH=-n, the TESTDATE information gets ignored. &DATEYYMD does not show the value in TESTDATE.

Instead it displays the current date. YRTHRESH is then an offset of the current date instead of the test date.

NOTE: SET YRTHRESH = -X also does not work with 3rd party software that changes the system date.

This reproduces under both MVS and VM environment.

Workaround:

N/A

Case Number: 32651044 Date Reported: 98/09/22

Summary: Use SCAN command to enter 2/29/2000: gets format error /c

Year 2000-Related Known Problems

Status: Submitted to Product Division

Product: FOCUS Op Sys: AIX Major Release: 685

Description:

A (FOC210) THE DATA VALUE HAS A FORMAT ERROR is received in SCAN FILE when entering 022900 into an MDY formatted field; 000229 into a YMD field; or 290200 into a DMY field.

Workaround:

Use MODIFY to enter the values into the database.

Case Number: 32661091 Date Reported: 98/09/23

Summary: MAINTAIN: TODAY does not return century

Status: Closed

Product: FOCUS Op Sys: ALL Major Release: 708

Description:

Using the TODAY function in Maintain never returns the century. You must then compute that value into an MDYY to get the current century. However, if you set TESTDATE into the year 2000, you still get 19 as the century instead of 20.

Workaround:

NONE

Year 2000-Related Known Problems

Case Number: 32721076 Date Reported: 98/09/29

Summary: &DATEMDYY, &DATEYYMD, &DATEDMY Y does not work with FMU files

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

New ¢ury dates will not display in a WINDOW TEXT DISPLAY.

This is &DATEYYMD, &DATEDMY Y, AND &DATEMDYY in 7.0.8R and 7.0.8. In 7.0.8, you can't use &YYMD, &DMYY OR &MDYY in addition to the &DATE variables.

Workaround:

For 7.0.8R, use &MDYY, &YYMD or &DMYY. There is no workaround for 7.0.8.

Case Number: 32781100 Date Reported: 98/10/05

Summary: Calling HLI subroutine causes YRTHRESH to be zero

Status: Submitted to Product Division

Product: FOCUS Op Sys: VM Major Release: 70

Description:

After calling an HLI subroutine, YRTHRESH is zero'ed. This occurs in 7.0.8R, and in 7.0.8 with the PTF 80088 (mega PTF) and PTF 93293 applied.

Year 2000-Related Known Problems

It does not occur in 7.0.8 if PTF 93293 is not applied.

Workaround:

N/A

Case Number: 32941102 Date Reported: 98/10/21

Summary: JUL DATE in packed format gives FOC1130 in rel 708 and 708R

Status: Closed

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

A packed field with a USAGE of JUL, and an actual of P4 gives a FOC1130 in release 7.0.8 if the field actually contains the century components.

This does not occur in release 7.0.7M, and is fixed in release 7.0.8R, if the field is described with a format of YYJUL. However, it is still a problem if the field is described with a format of JUL.

Another symptom of the same problem is a DEFINE:

```
xyz/jul = 1998001 (or '1998001' or '1998/001') GIVES FOC177  
(INVALID CONSTANT).
```

Workaround:

Create the field with a format of P7, and use GREGDT to put into a I8YYMD field. This can then be converted into a "smart date"

Year 2000-Related Known Problems

Case Number: 33011111 Date Reported: 98/10/28
Summary: Converting I4YM to YM with invalid data gives 0 in 7.0.8R
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Description:

In 7.0.8R, when converting I4YM or A4YM fields to YM format, FOCUS is giving blanks for the A4YM original format and giving the number of days from the base date in the I4YM original format.
These original dates are INVALID, but in 7.0.8, FOCUS adjusts the year and month.
This occurs in 7.0.8R on both MVS and VM.

Workaround:

Only use the A4YM format to a smartdate to produce a blank instead of the erroneous results given with the I4YM format conversion.

Case Number: 33071057 Date Reported: 98/11/03
Summary: A8YYMD to Smart Date, accepts bad data in year portion.
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Description:

Year 2000-Related Known Problems

An A8YYMD field with non-date values in the year portion of the incoming field is accepted into a Smart YYMD format field through a DEFINE, and will display an incorrect year.

Workaround:

Move the A8YYMD into an I8YYMD first. Then take the I8YYMD to move into the YYMD field.

Case Number: 33091037 Date Reported: 98/11/05

Summary: 7.0.8:COBOL/PLI pgm does not return century w/JULDAT/GREGDT

Status: Submitted to Product Division

Product: FOCUS Op Sys: MVS Major Release: 708

Description:

PLI or Cobol program that calls GREGDT or JULDAT returns 00 for century.

The correct century is returned in all releases except 7.0.8. This works also in 7.0.8R.

Workaround:

N/A

Year 2000-Related Known Problems

Case Number: 33231037 Date Reported: 98/11/19
Summary: Use of date subroutines in DM zero suppresses output
Status: Submitted to Product Division
Product: FOCUS Op Sys: AIX Major Release: 6.8

Description:

`-SET &DTE=AYMD(991231,1,'I6');`

returns an answer of 101, not 000101. This is standard behavior for DIALOGUE MANAGER, but causes a problem with Y2K implementation. MF fixes this via PTF for releases 7.0.8 and 7.0.8R, the same fix has to be requested for unix (LEADZERO=ON), for all Y2K products.

Workaround:

none

Case Number: 33481110 Date Reported: 98/12/14
Summary: DEFINE in MFD:fld = '&YYMD';\$ does not give date
Status: Submitted to Product Division
Product: FOCUS Op Sys: MVS Major Release: 708

Description:

DEFINEing a field to be &YYMD in the Master file Description will not give the value but uses the string instead.
This problem occurs in 7.0.8R and 7.0.9 Test.

Year 2000-Related Known Problems

This problem occurs in both VM and MVS.

Workaround:

Create the define in the FOCEXEC instead.

Case Number: 33491007 Date Reported: 98/12/15

Summary: Y2K:AYMD returns 0 in DM routine:subtract:cross century

Status: Submitted to Product Division

Product: FOCUS Op Sys: DYNIX Major Release: 655

Description:

The AYMD user-written subroutine, does not work when crossing the year 2000 mark by subtracting days from a date. In DIALOGUE MANAGER, 0 is returned. When using Defined fields, a blank is returned.

Adding days to cross from 1999 to 2000 works. Adding or subtracting days and staying in the century that was started from works.

Workaround:

Use smartdates in DEFINE fields.

Case Number: 23631017 Date Reported: 97/12/29

Summary: FORMAT YM:On PC displays Jan 01-31, 1901 as BLANKs

Year 2000-Related Known Problems

Status: Closed

Product: FOCUS for Windows Op Sys: WIN Major Release: 6.2

Description:

YM function used on the PC will display dates between Jan 01 - Jan 31, 1901 as blank dates. On another platforms e.g., EDA all Dates are displayed with the YM function.

Case Number: 32541006 Date Reported: 98/09/11

Summary: 621 RUNTIME Missing GRP Box.FCC

Status: Submitted to Product Division

Product: FOCUS for Windows Op Sys: WIN Major Release: 621

Description:

GRPBOX.FCC is not installed when installing RUNTIME 6.21

Workaround:

Copy GRPBOX.FCC from FOCUS\SYSTEM directory.

Case Number: 32591132 Date Reported: 98/09/16

Summary: AYMD Generates 00 for century marker

Status: Closed

Product: FOCUS for Windows Op Sys: WIN Major Release: 6.4

Year 2000-Related Known Problems

Description:

AYMD funtion generates 00 for century marker

Workaround:

None

Case Number: 32591133 Date Reported: 98/09/16

Summary: AYMD generates BLANK column

Status: Closed

Product: FOCUS for Windows Op Sys: WIN Major Release: 6.4

Description:

AYMD generates BLANK column

Workaround:

None

Case Number: 32591135 Date Reported: 98/09/16

Summary: DOC ENHANCEMENT: AYMD function

Status: Closed

Product: FOCUS for Windows Op Sys: WIN Major Release: 6.4

Description:

FFW6.4 help shows define field format in AYMD example to be I6YMD, this should be I8YMD.

Workaround:

```
DEFINE FILE EMPLOYEE
HIREDATE/YYMD=HIRE_DATE;
HDATE/I8YYMD=HIREDATE;
END
TABLE FILE EMPLOYEE
PRINT HIRE_DATE HDATE AND COMPUTE
AFTER35DAYS/I8YYMD = AYMD(HDATE, 35, AFTER35DAYS);
BY LAST_NAME BY FIRST_NAME
END
```

Case Number: 32741099 Date Reported: 98/10/01

Summary: FOC263: when using YYM function

Status: Closed

Product: FOCUS for Windows Op Sys: WIN Major Release: 6.4

Description:

FOC263 when attempting to YYM function. Creating Reports Manual states the following:

" The YM and YYM functions calculate the number of months that elapse between

Year 2000-Related Known Problems

two dates. " When using YYM you get a FOC263.

Workaround:

Use YM

Case Number: 32881124 Date Reported: 98/10/15

Summary: DIR command displays wrong creation date for Y2K date

Status: Closed

Product: FOCUS for Windows Op Sys: WIN/NT Major Release: 6.4

Description:

DIR in FFW displays 10-15-103, when the creation date of a file has a Y2K
Creation date. The date displays correct in Explorer.

Workaround:

None

Case Number: 33131073 Date Reported: 98/11/09

Summary: RSREADME.WRI:Is there such a file for 6.4 apart for 6.21

Status: Closed

Product: FOCUS for Windows Op Sys: WIN/NT Major Release: 6.4

Description:

In Report Server 6.40, when accessing the readme file, the 6.21 Report Server release readme comes up.

Workaround:

The 6.40 RS Readme file has been posted on the internet for public use:
FTP.IBI.COM/PUB/CSS/MICRO/RSERVER/RS64READ.zip

Case Number: 33351010 Date Reported: 98/12/01
Summary: FOC230 in intl 6.4 when © in TITLE of MASTER
Status: Submitted to Product Division
Product: FOCUS for Windows Op Sys: WIN Major Release: 640

Description:

When the character ^ is next to the single quote at the end of the TITLE section in a MFD a FOC230 error occurs, indicating a missing quote mark.

This is a problem with the Dutch release of 6.4 but not the US version.

Workaround:

Put a space between the special character and the single quote. This can be done on all the releases, but it is needed in the international versions.

Year 2000-Related Known Problems

Case Number: 32541098 Date Reported: 98/09/11

Summary: Site Analyzer:Won't print Year2000 reports OFFLINE-FOC034

Status: Submitted to Product Division

Product: SMART MODE Op Sys: VM Major Release: 708

Description:

Errors with offline to sysout due to size of printline.

Scenario:

1. Ex Sitesaf.
2. Choose 'Year 2000 Reports'.
3. Place the cursor next to 'Century and Year Threshold Report' and press pf5.
4. Change 'ONLINE' to 'OFFLINE' and press ENTER.

The report will not go offline. The following error message flashes by:

Smart/SAF - Report being generated offline ...

```
ERROR AT OR NEAR LINE 88 IN PROCEDURE SA200010FOCEXEC *  
(FOC034) THE PRINT LINE EXCEEDS THE MAXIMUM LINE WIDTH. WIDTH IS: 185
```

In addition, one is unable to view the report online and then print it.

Workaround:

If offline is sent to a file no error occurs.

Year 2000-Related Known Problems

Case Number: 32751071 Date Reported: 98/10/02

Summary: Site Analyzer: Unable to locate DBs on a sink machine

Status: Closed

Product: SMART MODE Op Sys: VM Major Release: 708

Description:

Site Analyzer will capture the location of a database when the user is attached to the mini disk. It does not capture the information when the database is located on a sink machine.

'Location not found' appears in the report.

Workaround:

NA

Year 2000-Related Known Problems